

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Une Proposition d'Algorithme pour le Placement Ergonomique des Objets Interactifs Concrets dans la Conception d'Interface Homme-Machine

Ouédraogo, Missiri; Ygueïtengar, Banta

Award date:
1993

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Une Proposition d'Algorithme
pour le Placement Ergonomique
des Objets Interactifs Concrets
dans la Conception d'Interface
Homme-Machine**

Mémoire présenté par :
Missiri Ouédraogo Banta Ygueïtengar

Facultés Universitaires Notre-Dame de la paix
Namur
Institut d'Informatique

**Une Proposition d'Algorithme pour le Placement
Ergonomique des Objets Interactifs Concrets dans la
Conception d'Interface Homme-Machine**

Mémoire présenté par Missiri Ouédraogo et Banta Ygueitengar
en vue de l'obtention du grade de Licencié et Maître en Informatique

Promoteur : Professeur François Bodart

Co-promoteur et Directeur de Mémoire :
Chercheur et Assistant Jean Vanderdonckt

Année académique 1992-93

Remerciements :

C'est pour nous un très grand plaisir de remercier toutes les personnes qui ont de près ou de loin contribué à la réalisation de ce travail.

Tout particulièrement, nous tenons à remercier :

Monsieur François Bodart, professeur à l'Institut d'Informatique, qui en a assuré la direction ;

Monsieur Jean Vanderdonckt, chercheur et assistant à l'Institut d'Informatique, qui nous a témoigné de son entière disponibilité et qui nous a donné de précieux conseils ;

Messieurs, Mesdames et Mademoiselle : Yabré, Aminata, Nadia, et Florent Ouédraogo, Djimrabaye Ngarboldoum, Doromaye Jacques, Massita Denise, Joseph Kémobé et Ndounangar Banta, pour leur soutien moral et leurs sacrifices.

RESUME

Ce mémoire examine le problème du placement automatique des objets interactifs concrets d'une interface homme-machine en se basant sur les règles ergonomiques. La première partie situe l'importance de l'ergonomie du placement automatique dans la génération d'interfaces. La deuxième partie étudie des outils traitant de ce problème. La troisième partie reprend plus en profondeur DON, un de ces outils, qui répond mieux à notre attente en raison de sa stratégie de placement plus évoluée. La quatrième partie propose des améliorations pour l'algorithme initial sur le placement dans TRIDENT (Tools foR Interactive Development ENvironment). Un exemple complet sert de fil conducteur à l'évaluation de DON et TRIDENT. Les propositions d'améliorations tiennent compte des solutions rencontrées dans l'étude des outils existants et des règles ergonomiques.

ABSTRACT

This study examines the problem of automatically placing concrete interaction objects in order to build a user-interface based on ergonomical rules. The first part shows how software ergonomics for automatic placement is important when generating user-interfaces. The second part analyzes tools solving this problem. The third part describes in further details DON, a more suited tool recognized for its well-behaved placement strategy. The fourth part suggests propositions for improving the initial placement algorithm in TRIDENT (Tools foR Interactive Development ENvironment). A complete example is equally used for evaluating both DON and TRIDENT. Improvements are taking into account several solutions addressed in the state of the art and the ergonomical rules.

TABLE DES MATIERES

Remerciements	
Résumé - Abstract	
Table des matières	
Introduction	1
Chap.I Introduction au problème du placement d'objets interactifs concrets.....	2
I.1 Définition des objets interactifs abstraits et des objets interactifs concrets	2
I.2 Sélection et placement des objets interactifs concrets.....	3
I.3 Ergonomie du placement.....	4
I.3.1 Ergonomie de la localisation	4
I.3.2 Ergonomie du dimensionnement	5
I.3.3 Ergonomie de l'arrangement.....	6
I.4 Conclusion.....	7
Chap.II Etude des outils existants en matière de génération automatique d'interfaces	8
II.1 Introduction.....	8
II.2 Etude des outils	9
II.2.1 ITS.....	9
II.2.2 JADE	12
II.2.3 SELECTORS.....	15
II.2.4 MacIDA.....	18
II.2.5 UIDE	21
II.2.6 HUMANOID	26
II.2.7 DON	31
II.2.8 GENIUS	37
II.3 Conclusion	40
Chap.III Le placement dans don	43
III.1 Introduction	43
III.2 Algorithme de placement.....	43
III.3 Quelques Aspects essentiels de la stratégie	46
III.3.1 Analyse des formes et des dimensions.....	46
III.3.2 Analyse de haut niveau.....	46
III.3.3 Contrôle des marges et des espacements.....	47
III.3.4 Utilisation de l'espace restant appelé "remainder area"	47
III.3.5 Les boutons OK et CANCEL.....	50
III.4 Application de l'algorithme à un exemple	52
III.5 Conclusion	59

Chap IV Placement des objets interactifs concrets dans TRIDENT	60
IV.1 Pré-requis	60
IV.2 Algorithme de placement.....	61
IV.2.1 Placement des OIC pour une Boîte de Dialogue Modale.....	61
IV.2.2 Caractérisation des Objets.....	62
IV.2.3 Fixation des Paramètres	63
IV.2.4 Calcul des Mesures Standards	64
IV.2.5 Calcul des Dimensions des Colonnes.....	70
IV.2.6 Calcul de la Proportion Intérieure.....	71
IV.2.7 Ajout des Boutons Standards	72
IV.2.8 Disposition des Boutons.....	72
IV.2.9 Choix des Marges	76
IV.3 Application de l'algorithme à un exemple.....	76
IV.3.1 Caractérisation des objets.....	76
IV.3.2 Fixation des paramètres.....	77
IV.3.3 Calcul des mesures standards	78
IV.3.4 Calcul de Z	81
IV.3.5 Calcul des dimensions des colonnes.....	81
IV.3.6 Ajout des boutons standards	82
IV.4 Comparaison des Algorithmes DON et TRIDENT	83
IV.4.1 Souplesse des deux stratégies.....	83
IV.4.2 Analyse des formes et des dimensions	84
IV.4.3 Solutions d'overflow	84
IV.4.4 Placement des boutons.....	85
IV.4.5 Placement en deux colonnes.....	85
IV.4.6 Proportion interne.....	86
IV.5 Critiques et améliorations de l'algorithme.....	86
IV.5.1 Modifications indispensables	86
IV.5.2 Quelques modifications préférables	88
IV.5.3 Autres remarques et suggestions	89
IV.5.4 Stratégie "Droite-Dessous" à un niveau.....	92
IV.5.5 Arrangement des items de boutons-radio.....	93
IV.6 Conclusion	103
Ch. V Conclusion générale	104
Bibliographie	106

INTRODUCTION

Le sujet de ce mémoire est consacré au problème du placement automatique des objets interactifs concrets constituant une interface homme-machine. Le mémoire s'inscrit dans la continuité des recherches effectuées dans le cadre du projet TRIDENT relatif aux applications de gestion hautement interactives. Dans ce projet, l'Interface Homme-Machine prend une part beaucoup plus importante que par le passé, notamment par le respect des règles ergonomiques.

CHAPITRE I

INTRODUCTION AU PROBLEME DU PLACEMENT D'OBJETS INTERACTIFS CONCRETS

I.1 DEFINITION DES OBJETS INTERACTIFS ABSTRAITS ET DES OBJETS INTERACTIFS CONCRETS

"Un objet interactif concret (en abrégé, OIC) constitue, au sens large du terme, un objet de dialogue utilisé pour la saisie et/ou l'affichage d'informations relatives à la tâche de l'opérateur" (B. Sacré, I. Sacré-Provot et J. Vanderdonckt).

Exemples :

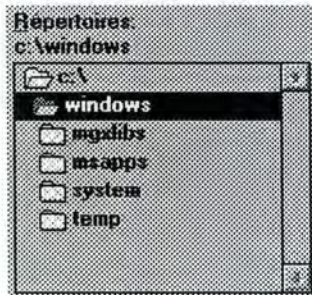
1. une liste de combinaison déroulante



2. un bouton de commande

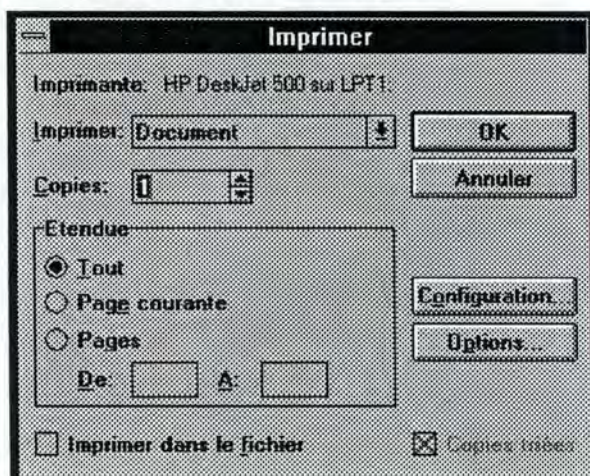


3. une liste de sélection



Un objet interactif concret peut être soit simple (par exemple : un bouton de commande, un boîte à cocher) soit composite (par exemple, une boîte de dialogue). Le dernier peut être lui-même composé d'objets simples ou composites (par exemple, une boîte de regroupement peut entourer un bouton radio à trois items et deux champs d'édition, comme dans l'exemple ci-après).

Exemple : une boîte de dialogue



Le concept d'objet interactif concret pose un problème de généralisation : un même objet d'interaction peut se retrouver dans un autre environnement physique sous une autre appellation, avec une autre représentation graphique et, quelquefois, avec une variante dans le changement de présentation.

Exemple :

Style :	<input checked="" type="checkbox"/> Bold	Style :	<input checked="" type="checkbox"/> Bold
	<input type="checkbox"/> <i>Italic</i>		<input type="checkbox"/> <i>Italic</i>
	<input checked="" type="checkbox"/> <u>Underlined</u>		<input checked="" type="checkbox"/> <u>Underlined</u>
	<input type="checkbox"/> Outlined		<input type="checkbox"/> Outlined

Style: **Bold** *Italic* Underlined **Outlined**

Pour résoudre ce problème, on introduit la notion d'objet interactif abstrait.

"Un objet interactif abstrait (en abrégé, OIA) est une abstraction de l'ensemble des objets interactifs de même type indépendamment des environnements physiques qui l'accueillent" (B. Sacré, I. Sacré-Provot, J. Vanderdonckt).

Il découle de cette définition qu'un OIA n'a pas de représentation graphique et que, dans un environnement particulier, un OIA existe ou n'existe pas qu'à travers sa présence ou son absence de concrétisation en OIC au sein de l'environnement considéré.

I.2 SELECTION ET PLACEMENT DES OBJETS INTERACTIFS CONCRETS

Dans le pré-requis, nous avons montré comment les OIA étaient obtenus à partir de la sémantique de l'application. Les OIA sont structurés de façon hiérarchique pour tenir compte notamment de leur succession logique dans l'interface. La hiérarchie des OIA est ensuite transformée en une hiérarchie des OIC en choisissant un environnement cible. Ces OIC sont alors sélectionnés et placés dans l'interface en respectant les règles ergonomiques du placement. Ces règles ergonomiques prennent en compte les trois parties principales du placement qui sont : la localisation, le dimensionnement et l'arrangement.

La localisation s'intéresse au positionnement logique de l'OIC sur l'écran. Elle prend donc en compte la position, l'alignement et la justification des OIC entre eux.

Le dimensionnement s'occupe de l'uniformisation et de la standardisation de la taille des OIC. Il prend en compte par exemple l'uniformisation des abréviations, le nombre maximum de caractères pour les libellés, la taille d'un champ d'édition ou d'une boîte de dialogue, l'harmonisation de la longueur et de la hauteur d'une boîte de dialogue, etc.

L'arrangement tient compte de l'orientation des OIC (position sur l'espace d'affichage) et de certaines contraintes telles que la succession logique entre des OIC, leur ordonnancement, l'équilibrage des OIC sur l'espace d'affichage, Ainsi nous aurons par exemple une boîte de regroupement pour des objets associés à des données logiquement liées, un balancement de l'affichage de manière symétrique, de gauche à droite, de bas en haut, etc.

I.3 ERGONOMIE DU PLACEMENT

Nous donnons à l'ergonomie de l'interface utilisateur la définition suivante :

"Tout ce qui, dans un système informatique, influence la participation de l'utilisateur à des tâches informatisées. Elle détermine ainsi :

- la facilité et l'efficacité d'utilisation,
- la facilité d'apprentissage, et en conséquence :
- l'efficacité et la productivité de l'utilisateur au travail" (J.P. Meinadier).

L'ergonomie de l'interface utilisateur prendra donc en compte aussi bien les aspects tels que :

- moyens d'interaction et types d'interaction utilisés,
- choix des OIC pour saisir et/ou présenter les données et/ou résultats,
- localisation, dimensionnement et arrangement des OIC

que les aspects tels que :

- structure du dialogue entre l'application et l'utilisateur,
- information à présenter dans chacune des phases du dialogue,
- guidance (ou assistance) de l'utilisateur.

Nous ne nous intéressons ici qu'aux aspects localisation, dimensionnement et arrangement des OIC. Un ensemble exhaustif de règles pour un placement ergonomique des OIC pour des applications interactives est donné dans [VAN93a]. Nous ne reprenons ici que les aspects essentiels.

I.3.1 ERGONOMIE DE LA LOCALISATION

Pour accroître l'efficacité d'utilisation et la facilité d'apprentissage de l'utilisateur, le placement des OIC doit s'inspirer des orientations suivantes.

La cohérence

Il s'agit ici d'aider l'utilisateur à acquérir une connaissance motrice qui lui permette d'anticiper des actions physiques : il peut déplacer à l'avance le pointeur sur un OIC à modifier, il peut à l'avance focaliser son attention vers le point de l'écran contenant la donnée cherchée. On s'efforcera donc d'aider l'utilisateur à se construire mentalement un schéma pour chacun des types d'OIC qu'il manipule. A cette fin :

- suivre un standard d'affichage, si possible ;
- la position des OIC doit être cohérente avec les conventions courantes de l'utilisateur ;
- la position des OIC doit être cohérente à travers toute l'application, et si possible à travers toutes les applications du poste de travail ;
- tous les OIC ayant un même centre d'intérêt doivent être positionnés proches les uns des autres (et proches de ce centre d'intérêt).

La succession des OIC

Ici également, il s'agit d'aider l'utilisateur lors de la recherche d'une information particulière. A cette fin :

- les objets qui normalement doivent être manipulés en premier (respectivement, en dernier) doivent être placés à gauche et en haut (respectivement, en bas à droite) de l'écran ;
- les OIC systématiquement utilisés lors d'un dialogue doivent être placés en premier ;
- les données les plus importantes ou à retenir doivent être présentées en premier.

L'utilisation de l'écran

De ce point de vue, il s'agira d'accroître la lisibilité et l'esthétique des écrans générés. Pour cela :

- on tentera d'avoir une répartition homogène des OIC dans une fenêtre, c'est-à-dire qu'on tentera d'utiliser de façon équitable les 4 quadrants de la fenêtre ;
- si possible, la fenêtre devra être placée au centre de l'écran ;
- tout affichage devra comporter un titre ou un en-tête décrivant son contenu ou son objectif, placé sur sa première ligne et clairement séparé du reste de l'affichage ;
- les marges, les interlignes, les intercolonnes doivent être suffisantes pour accroître la lisibilité de l'affichage et l'isolement des sous-groupes d'OIC lorsque la fenêtre en comporte plusieurs ; des séparateurs et des boîtes de regroupement peuvent être utilisés pour améliorer l'isolement des sous-groupes.

I.3.2 ERGONOMIE DU DIMENSIONNEMENT

Du point de vue des dimensions des OIC, on cherchera surtout à ce que la taille de ceux-ci permette qu'on les atteigne de façon aisée et que leur forme soit esthétiquement harmonieuse. On pourra à cette fin s'inspirer des recommandations suivantes :

- les OIC ne doivent pas être de trop petites tailles ;
- éviter les lignes d'écran trop longues ;
- pas de barres de menus de plus d'une ligne ;
- dans un OIC composé, éviter d'avoir plus de deux colonnes et éviter des colonnes de plus de 35 caractères ;
- il est souhaitable que les boutons arrangés verticalement aient des longueurs égales et que ceux arrangés horizontalement aient des hauteurs égales ;
- éviter de disposer plus de 7 boutons dans une même boîte de dialogue ;
- les champs d'édition pour données de longueurs courtes ou de longueurs fixes doivent avoir les mêmes longueurs que ces données ; si la longueur d'un champ d'édition doit dépasser 40, le partitionner si possible ;
- ne pas afficher plus de 9 items dans une liste de sélection ou de combinaison ; ce nombre doit être revu à la baisse pour un utilisateur novice ;
- la taille d'une boîte de dialogue doit être déterminée par celles des OIC qu'elle contient ;
- les proportions d'une fenêtre ou d'une boîte de dialogue doivent être harmonieuses.

I.3.3 ERGONOMIE DE L'ARRANGEMENT

Ici, on cherchera à organiser l'écran de telle sorte que l'utilisateur n'ait aucune peine à repérer les différents groupes d'objets composant la fenêtre ou la boîte de dialogue. On cherchera même à ce que, très rapidement, il puisse anticiper des actions physiques en vue de manipuler un objet particulier de l'écran. On se souviendra surtout que :

l'arrangement doit être autant que possible logique

- L'arrangement des OIC qui est le plus significatif pour l'utilisateur est celui qui correspond à la manière dont il exécute les actions pour accomplir la tâche ;
- l'arrangement des items doit être suivant l'ordre de pensée de l'utilisateur, suivant l'ordre de survenance, suivant l'ordre voulu par les utilisateurs ou suivant l'ordre de compréhension ;
- les OIC associés à des données logiquement liées doivent être groupés ;
- les OIC de contrôle qui sont sémantiquement liés doivent voir ce lien représenté au moyen de séparateurs, de boîtes de regroupement et d'espaces blancs ;
- l'arrangement des données affichées doit suivre une découpe en zones fonctionnelles (par exemple : chaque type d'information doit être affiché dans une zone spéciale cohérente toujours localisée au même endroit) ;
- éviter de délimiter un groupe de boutons de commande ou des listes de sélection et de combinaison non déroulantes par une boîte de regroupement ;
- éviter de disposer un autre objet interactif dans un bouton de commande.

l'arrangement doit apporter un renfort visuel

L'arrangement des données devrait montrer visuellement leur structure.
Les zones fonctionnelles devraient être prédictibles.

l'arrangement doit être esthétique

La présentation d'une boîte de dialogue doit suivre une grille virtuelle qui garantisse un minimum d'esthétique. Cette grille doit gouverner le placement de tous les OIC principaux.

Placer les OIC de petite taille loin de l'axe de symétrie vertical de l'OIC composé, et placer les OIC de grande taille près de l'axe de symétrie vertical de l'OIC composé

l'arrangement doit minimiser le mouvement oculaire

Les OIC critiques doivent autant que possible être disposés le long de la ligne de balayage oculaire en Z.

La densité d'affichage des données ne doit pas être exagérée. A titre indicatif, nous proposons le tableau de densités suivant :

- 18% : écrans subjectivement jugés acceptables,
- 27% : densité recommandée,
- 40% : page papier bien conçue,
- 50% : densité maximale perçue et recommandée,
- 60% : le temps de recherche grandit drastiquement à partir de cette densité,
- 80% : densité de la plupart des écrans d'ordinateur.

La densité d'affichage globale ne doit pas franchir les 30%. Cependant, les données textuelles destinées à la lecture continue peuvent être condensées.

l'arrangement doit minimiser le mouvement manuel

L'arrangement devrait permettre de réduire à son minimum le mouvement de la main pendant l'exécution d'une tâche.

l'arrangement doit être standardisé, si possible

Chaque fois que possible, arranger les OIC composés suivant des standards existants.

I.4 CONCLUSION

Comme nous l'avons vu dans l'ergonomie du placement, une interface esthétique ne peut être obtenue qu'en respectant un certain nombre de règles ergonomiques tant sur la localisation, le dimensionnement que sur l'arrangement. Pour mieux comprendre l'importance de ces trois points, nous étudierons dans le chapitre II un certain nombre d'outils qui prennent en compte le placement des OIC. Nous consacrerons un chapitre particulier (le chapitre III) à DON, un outil qui s'occupe de la localisation, du dimensionnement et de l'arrangement. Nous aborderons ensuite au chapitre IV le placement des OIC dans TRIDENT.

Une comparaison est faite entre les deux algorithmes DON et TRIDENT. En s'appuyant sur un exemple traité suivant les deux algorithmes, nous formulerons des critiques en vue d'améliorer l'algorithme TRIDENT.

Le dernier chapitre tire les conclusions générales sur le placement.

CHAPITRE II

ETUDE DES OUTILS EXISTANTS EN MATIERE DE GENERATION AUTOMATIQUE D'INTERFACES

II.1 INTRODUCTION

Plusieurs outils existent en matière d'aide à la génération automatique d'interfaces. Nous nous proposons dans ce chapitre d'en étudier quelques-uns parmi les plus évolués. Nous les analyserons suivant la grille ci-après :

- 1- nom général du système : nom sous lequel est connu le système ;
- 2- type général du système : type d'environnement fourni par le système ; par exemple : environnement complet, UIMS (User Interface Management System), générateur, toolkit, ...
- 3- auteur(s) ;
- 4- année de présentation : année à laquelle l'outil a été présenté pour la première fois ;
- 5- référence(s) bibliographique(s) : document(s) effectivement consulté(s) pour l'analyse ;
- 6- liste et définition des objets composant l'outil : définition des parties essentielles de l'outil ; par exemple : éditeur de spécification, éditeur de présentation, base de règles, base de connaissances, ...
- 7- schéma de l'architecture générale : diagramme de flux simplifié qui donne la structure du fonctionnement de l'outil et la procédure à suivre ;
- 8- source(s) de la génération : éléments de spécification sur lesquels l'outil se base pour démarrer le processus de génération ;
- 9- produit(s) de la génération : objets interactifs concrets que l'outil peut générer ;
- 10- type et langage de spécification ; par exemple : langage déclaratif, langage procédural, IDL (Interface Description Language), UIL (User Interface Language), ...
- 11- phase de sélection d'OIA : description de la phase de sélection d'objets interactifs abstraits si elle existe dans la stratégie de l'outil et si les références bibliographiques le permettent ;
- 12- phase de placement d'OIC : description de la phase de placement d'objets interactifs concrets si elle existe dans la stratégie de l'outil et si les références bibliographiques le permettent ;
- 13- type de règles utilisées : les règles utilisées sont-elles modifiables ou non, visibles ou non, et explicites ou non ;
- 14- forme des règles ergonomiques ; par exemple : fichier texte, base de connaissances, base de règles, réseau sémantique, liste de "si ... alors ... sinon", table de décision.

Les 14 points seront étudiés pour l'outil le cas échéant. Par exemple, tous les outils ne font pas apparaître une phase de sélection d'OIA. Pourtant nous tenterons dans la mesure du possible d'en identifier les contours.

II.2 ETUDE DES OUTILS

II.2.1 ITS

Nom général du système

ITS (Interactive Transactive System)

Type général du système

Générateur de vues

Auteurs

Wiecha C., Bennett W., Boies S. et Gould J.

Année de présentation

1989

Référence bibliographique

Wiecha C., Bennett W., Boies S. et Gould J., "*Generating Highly Interactive User Interfaces*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'89 (Austin, mai 1989), pp. 277-282.

Liste et définition des objets de l'environnement

- Le *Compilateur de Dialogues* :

Il utilise les types de données, les tables des données partagées entre l'application et l'interface et les fichiers de dialogue et fournit un graphe d'analyse.

- Le *Compilateur de Styles* :

Il reçoit un fichier de règles de style et le graphe d'analyse généré par le compilateur de dialogues et attache des techniques d'interaction et des vues provenant d'une boîte à chacun des noeuds du graphe d'analyse.

- Le *Gestionnaire de Dialogue* :

C'est un éditeur de présentation à manipulation non directe. Il reçoit la version finale du graphe d'analyse, exécute les actions (non graphiques) du programme et fait appel à un gestionnaire de vues (gestionnaire d'écran) pour afficher les vues définies par le programmeur de style.

- Le *Gestionnaire d'écran*.

- Le *Fichier de Règles de Styles* :

Il contient des règles écrites par l'expert en style. Ces règles permettent le choix des objets qui interviendront dans les vues.

- *Le Fichier de Dialogue* :

Il contient une hiérarchie d'étiquettes telles que forme, liste et choix. Chaque étiquette est associée à des routines prédéfinies après exécution de règles écrites dans un style particulier.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Définition de types de données,
- ☐ Hiérarchie d'étiquettes de dialogue

Produits de la génération

- ☐ Actions
- ☐ Vues

Type et langage de spécification

- ☐ Déclarations textuelles
 - de types de données
 - de tables
 - de listes
 - de trames
- ☐ Définitions textuelles des vues.

Phase de sélection d'objets interactifs abstraits

- ☐ Présente.
- ☐ Suite de tests, en utilisant le fichier des règles de style.

Phase de placement d'objets interactifs concrets

- ☐ Absente (laissée à la responsabilité du concepteur)

Types de règles utilisées

- ☐ Règles de dimensionnement et d'arrangement élémentaires à manipulation directe,
- ☐ Réutilisables et partageables,
- ☐ Modifiables avec recompilation.

Forme des règles ergonomiques

- ☐ Pas de règles ergonomiques.

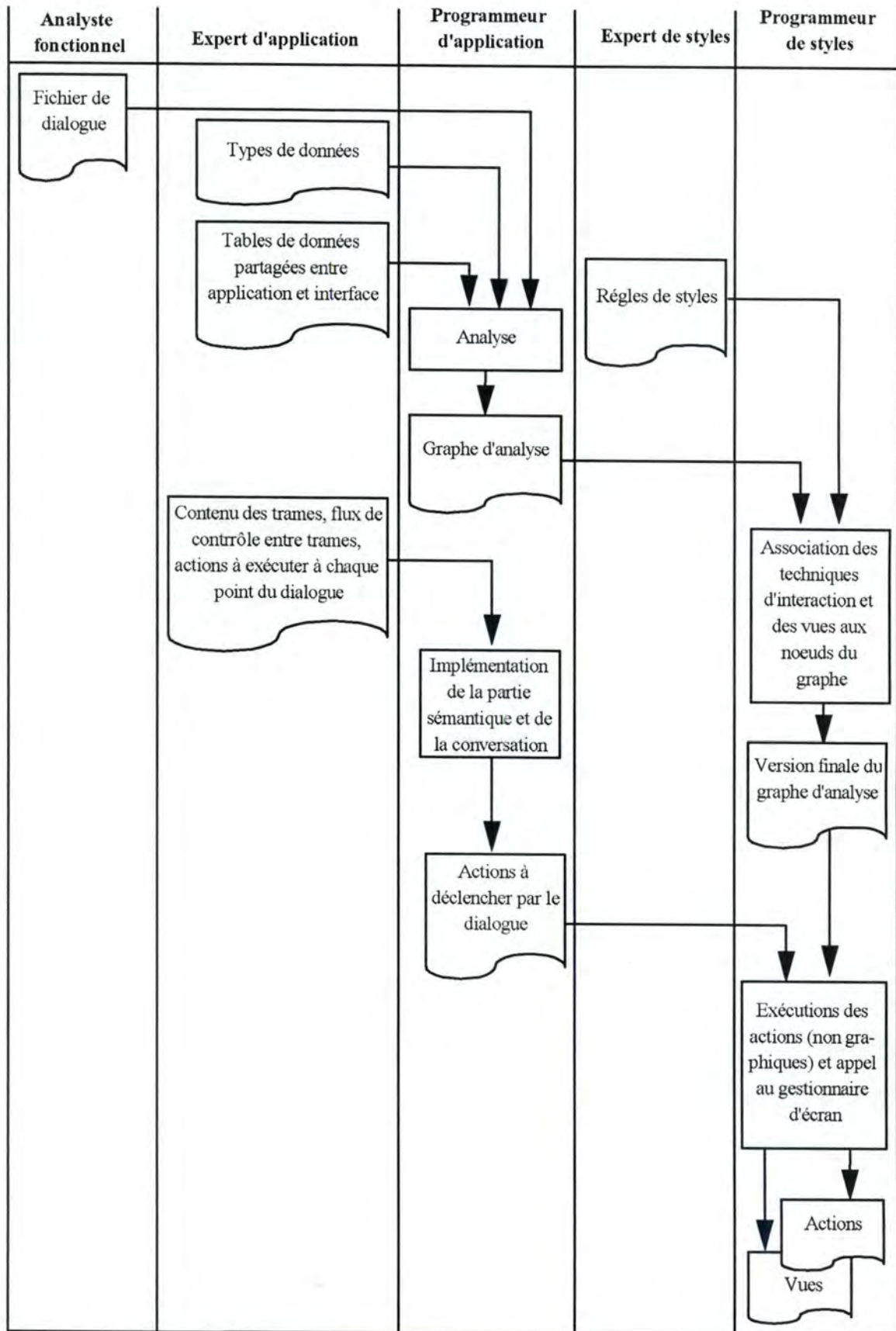


Figure 2.1 : Schéma de l'architecture générale de ITS

II.2.2 JADE

Nom général du système

JADE (Judgement-based Automatic Dialog Editor)

Type général du système

Générateur automatique de Dialogue graphique.

Auteurs

Vander Zanden Brad et Myers Brad A.

Année de présentation

1990

Référence bibliographique

Vander Zanden B. et Myers, B. A., "*Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'90, (Seattle, avril 1990), pp. 27-34.

Liste et définition des objets de l'environnement

- La Base Principale de Règles :

Elle contient l'ensemble des règles qui permettent de définir les propriétés des objets interactifs concrets.

- L'Editeur de Règles de Jade :

C'est un éditeur à manipulation directe qui permet d'éditer de nouvelles règles, qui peuvent être ajoutées à la base principale de règles, à la base de règles et graphiques d'exception ou à la base de règles de comportement et présentation. Il permet également d'éditer une spécification textuelle de ces règles.

- La Base de Règles de Comportement et Présentation :

Elle contient les informations relatives au positionnement des objets interactifs concrets contenus dans la base de données de comportement et présentation graphiques.

- La Base de Données de Comportement et Présentation graphiques :

Elle contient la définition des objets interactifs concrets

- Jade :

Il utilise en entrée le fichier de spécification du dialogue, la base de règles de comportement et de présentation, la base de données de comportement et présentation graphiques et le fichier de règles et graphiques d'exception.

Il affiche les objets interactifs concrets

- Le Fichier de Règles et Graphiques d'Exception :

C'est un fichier optionnel qui contient des règles et des graphiques qui complètent ou modifient les règles et graphiques de base. Ce fichier n'est utilisé que pour l'application pour laquelle il a été défini.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Spécification textuelle de la sémantique

Produits de la génération

- ☐ boîtes de dialogue,
- ☐ menus.

Type et langage de spécification

- ☐ Langage textuel et déclaratif.

Phase de sélection d'objets interactifs abstraits

- ☐ Absente.

Phase de placement d'objets interactifs concrets

- ☐ Il existe des règles de dimensionnement et d'arrangement.
- ☐ On peut, par exemple, écrire des règles
 - pour placer des objets, déterminant quand revenir à la ligne,
 - pour spécifier les offsets (décalages) horizontaux et verticaux entre objets
 - et pour spécifier quel police de caractères utiliser.

Exemple :

```
( at-right-rule
  ( ( :left (formula (+ (gv obj :right) offset) ) )
    ( :top (formula (gv obj :top) ) ) )
  obj offset )
```

Cette règle positionne un objet à droite d'un objet de référence. La formule correspondant à :left trouve la marge droite de l'objet de référence et ajoute à cette marge un nombre de pixels égal à *offset*. La formule correspondant à :top trouve la marge de haut de l'objet de référence.

Types de règles utilisées

- ☐ Modifiables, visibles et explicites.

Forme des règles ergonomiques

- ☐ Bases de règles.

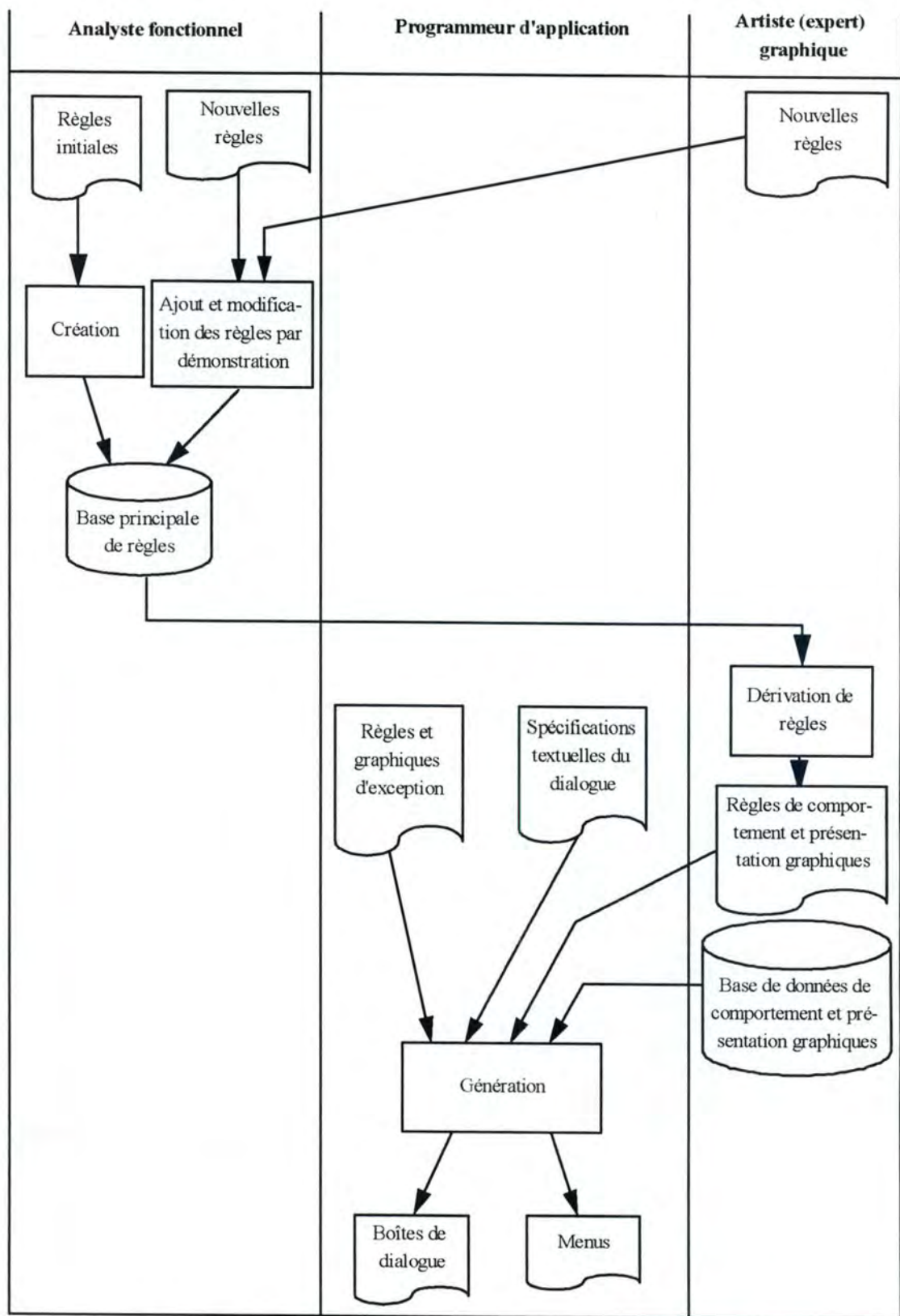


Figure 2.2 : Schéma de l'architecture générale de JADE

II.2.3 SELECTORS

Nom général du système

Selectors

Type général du système

Bibliothèque de procédures : "selectors" fondés sur le toolkit.

Auteur

Johnson, J.

Année de présentation

1992

Référence bibliographique

Johnson, J., "*Selectors : Going beyond User-Interface Widgets*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'92, (Monterey, 3-7 mai 1992), pp. 273-279.

Liste et définition des objets de l'environnement

- *Data Selectors* :

Un "data selector" est soit un objet sémantique de base soit un objet de choix.

Un objet sémantique de base permet de représenter des valeurs de données simples et couramment utilisées. Exemples : les nombres, les valeurs de vérité, les couleurs, l'heure, les montants en dollars. On dispose d'une collection de représentations de ces objets de base, chaque objet pouvant être représenté de plusieurs façons différentes.

Un objet de choix encapsule la sémantique d'un type particulier de choix, indépendamment de l'interface utilisé pour faire ce choix. Il est défini en termes d'objets sémantiques de base. On distingue 4 types de choix possibles :

- sélection d'une valeur unique dans un petit ensemble de valeurs discrètes connues (exemple : voter pour une personne parmi les candidats suivants : Dupont, Dupond, Durant, Durand, Paul, Jean) ; un cas particulier d'une telle sélection est la sélection d'une valeur parmi deux valeurs opposées ;
- sélection de plusieurs valeurs dans un ensemble de valeurs discrètes (exemple : voter pour trois personnes parmi les candidats suivants : Dupont, Dupond, Durant, Durand, Paul, Jean) ;
- sélection d'une valeur unique dans un domaine potentiellement grand de valeurs non discrètes (exemple : Volume : 0 - 1000) ;
- sélection de plusieurs valeurs ou sous-domaines dans un ensemble grand et ordonné (exemple : périodes où l'infirmière était avec le patient le 6 mai : 00:01 - 24:00).

On dispose d'une collection de représentations des objets de choix, chaque objet de choix pouvant être représenté de plusieurs façons différentes (exemple : un choix unique dans un petit ensemble de valeurs discrètes peut être représenté comme un bouton-radio,

un ensemble de boîtes à cocher, un bouton cyclique, un champ d'édition ou comme un menu déroulant).

Les "data selectors" peuvent être définis :

- en extension ; par exemple : {rouge, vert, bleu} ;
- en compréhension ; par exemple : [0...50] ;
- ou à l'aide de formules logiques ; par exemple : Odd(n).

- *Command Selectors* :

Un "command selector" est un objet sémantique qui permet d'invoquer un groupe quelconque de commandes. Il a un domaine spécifié par un "command list". Il a aussi une valeur qui indique laquelle des commandes est active, un nombre minimum et un nombre maximum qui délimitent le nombre de commandes pouvant être actives à un moment donné. On dispose d'une collection de représentations pour les "command selectors".

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Pas de génération

Produits de la génération

- ☐ Pas de génération, mais on obtient des boîtes de dialogue complètes.

Type et langage de spécification

- ☐ Pas de langage de spécification

Phase de sélection d'objets interactifs abstraits

- ☐ Sélection par association (détermination du "selector" à partir de la sémantique de l'application).
- ☐ Règles de sélection imbriquée dans le code du programme.

Phase de placement d'objets interactifs concrets

- ☐ Présente, mais non décrite dans l'article.

Types de règles utilisées

- ☐ Visibles et modifiables.

Forme des règles ergonomiques

- ☐ Ensemble d'associations.

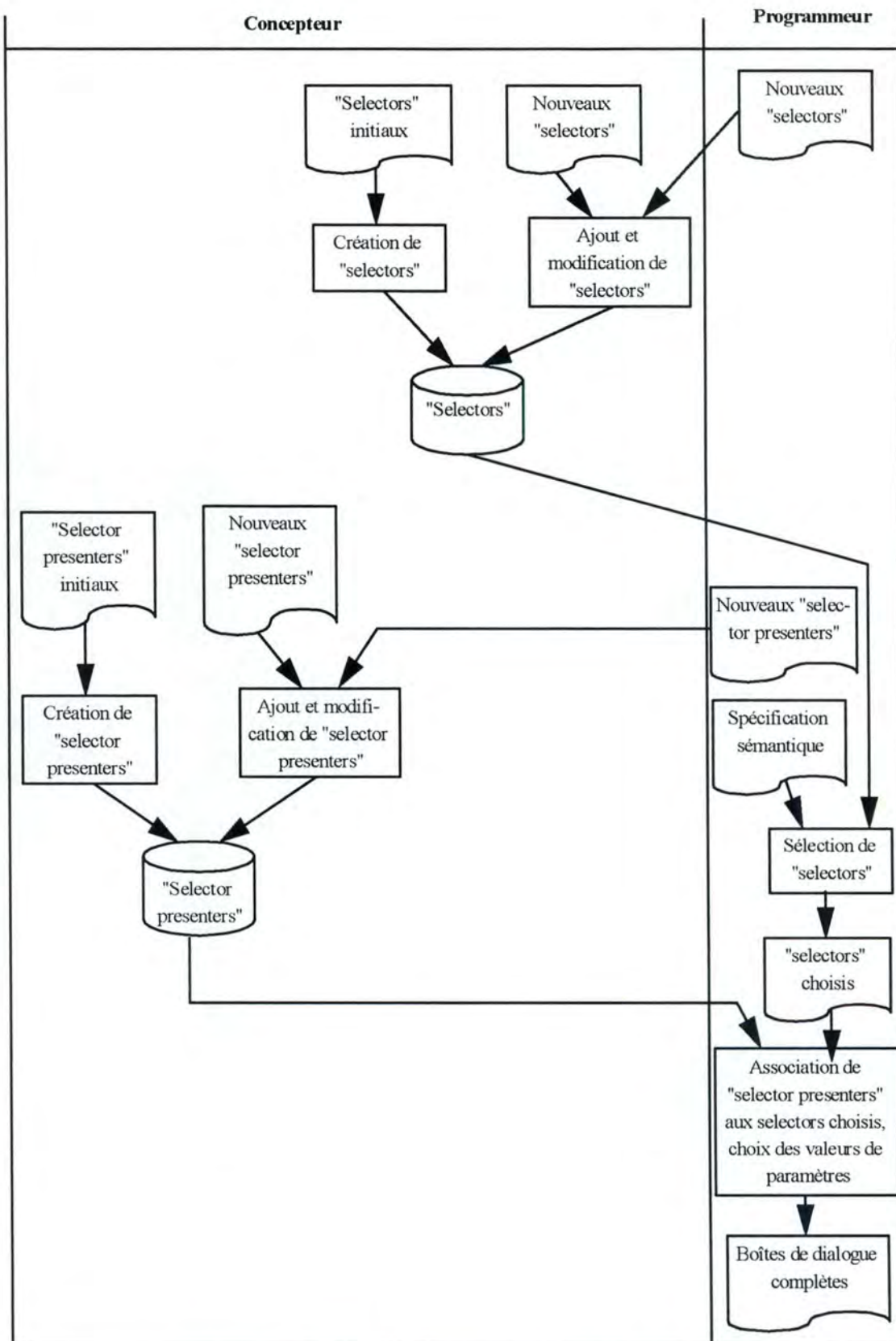


Figure 2.3 : Schéma de l'architecture générale de SELECTORS

II.2.4 MACIDA

Nom général du système

MacIda

Type général du système

Générateur d'interface

Auteurs

Petoud, I. et Pigneur, Y.

Année de présentation

1989

Références bibliographiques

Petoud, I. et Pigneur, Y., "An Automatic and Visual Approach for User Interface Design" in IFIP Working Conference Engineering for Human Computer Interaction, (Napa Valley, août 1989), pp. 7-14.

Petoud, I. "Génération Automatique de l'Interface Homme-Machine d'une Application de Gestion hautement interactive", Inf. Chabloz, Tolochenaz, 1990, pp.171-192.

Larson, J.A., "Interactive Software : Tools for Building Interactive User Interfaces", Englewood Cliffs, NY : Yourdan Press, 1992, pp. 298-300.

Liste et définition des objets de l'environnement

- Le Contrôleur de Présentation :

Il permet l'affichage et la visualisation des informations à l'écran. Il interprète les actions de l'utilisateur au clavier ou avec la souris. Il permet également de communiquer des informations ou actions de l'utilisateur au gestionnaire de la conversation.

Il est implémenté au-dessus de MacApp, un outil Macintosh permettant la création d'application par utilisation de classes d'objets.

- Le Gestionnaire de conversation :

Il crée une structure de données qui représente le graphe d'enchaînement des fonctions. Il réalise le lien entre la présentation et la conversation en mettant en relation les objets interactifs de saisie avec le graphe. Il guide l'utilisateur pendant l'introduction des données. En cas d'erreur, il affiche un message d'erreur et met à la disposition de l'utilisateur un panneau de contrôle (ou tableau de bord) et un graphe d'enchaînement de fonctions. Le graphe montre les fonctions activables et celles qui ne le sont pas parce que toutes leurs données d'entrée ne sont pas encore introduites.

- Le Générateur d'Interface :

Il produit toutes les déclarations permettant d'obtenir les OIC nécessaires au fonctionnement de l'interface.

A chaque type d'entité ou type d'association il associe une fenêtre. La fenêtre porte le nom du type correspondant. A chaque fonction sémantique de l'application (exemples : validation d'un client, ouverture d'un fichier, ...), il associe un bouton. Le bouton porte le nom de la fonction.

Il génère une boîte pour chaque message d'erreur.

Il fixe les paramètres de présentation, de localisation et de dimensionnement élémentaires par défaut.

Il produit le code du programme exprimé en termes d'objets de MacApp.

- L'Editeur d'Interface Utilisateur :

C'est un éditeur à manipulation directe qui permet de personnaliser l'interface générée automatiquement par l'outil. Cette personnalisation ne doit pas modifier les données de la spécification fonctionnelle. Cette responsabilité incombe à l'utilisateur ; il n'y a pas de garantie de la complétude et de la cohérence parce que l'utilisateur est libre de faire ce qu'il veut.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Spécification textuelle des fonctions et des données de l'application, en IDA/ISL
- ☐ Raffinement par manipulation directe

Produits de la génération

- ☐ Fenêtres
- ☐ Boîtes d'édition
- ☐ Menus déroulants
- ☐ Menus en sur-affichage
- ☐ Boutons de commande
- ☐ Boîtes de dialogue
- ☐ Tables

Type et langage de spécification

- ☐ Langage de spécification textuelle : IDA/ISL

Phase de sélection d'objets interactifs abstraits

- ☐ Non présente.

Phase de Sélection d'Objets Interactifs Concrets

- ☐ Présente.
- ☐ Pour chaque entité, on sélectionne une fenêtre.
- ☐ Pour chaque attribut d'une entité, on sélectionne un champ d'édition, qui est toujours unilinéaire.

- Pour chaque agrégat répétitif, on sélectionne une table.
- Pour toute fonction, on sélectionne un bouton de commande.

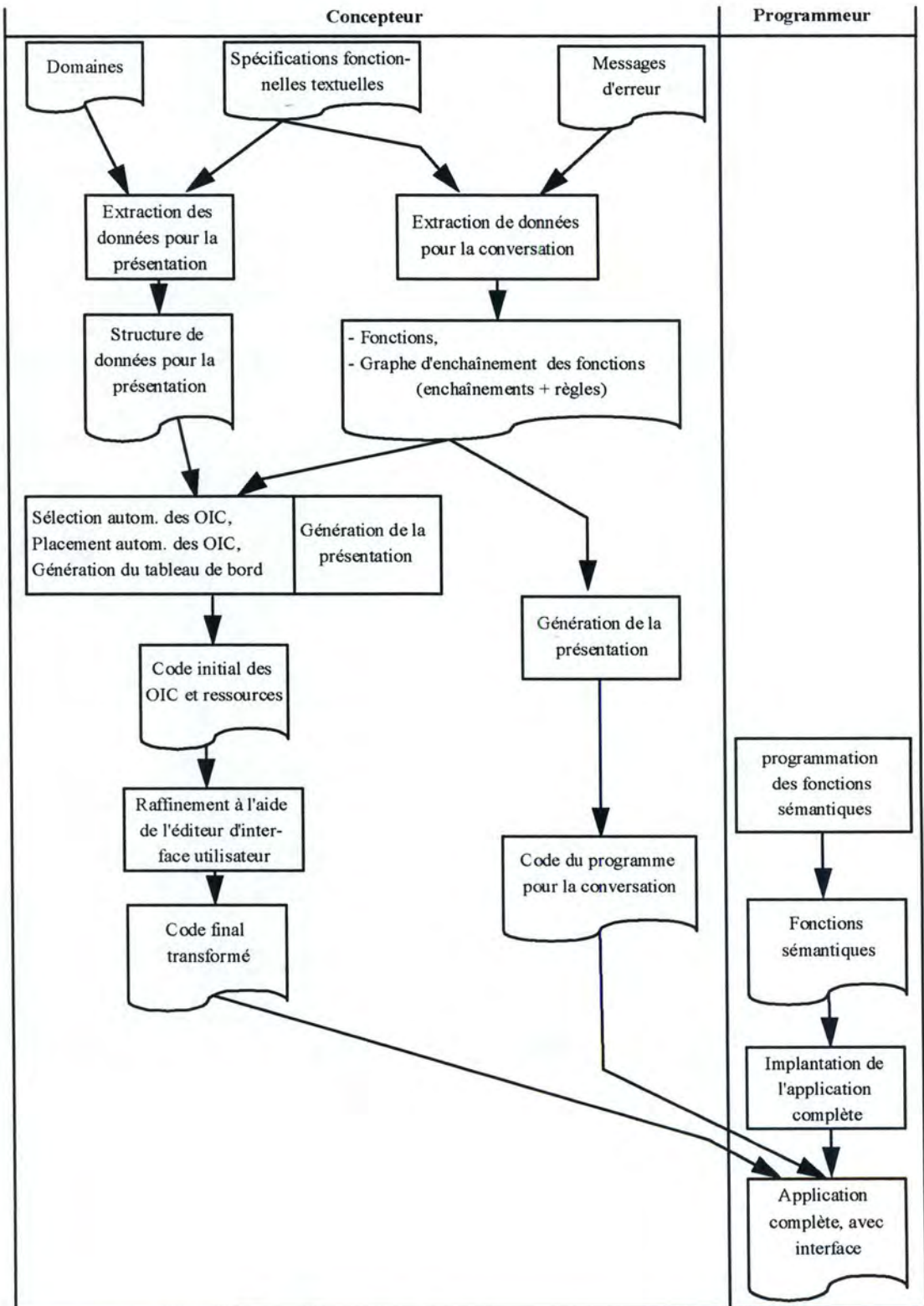


Figure 2.4 : Schéma de l'architecture générale de MacIDA

Phase de placement d'objets interactifs concrets

- ☐ Présente.
- ☐ Les fenêtres représentant les types d'entité et les types d'association sont positionnées de telle sorte que leur emplacement rappelle le schéma E/A du concepteur.
- ☐ Les champs représentant les attributs des types d'entité et des types d'association sont placés en colonne dans les fenêtres suivant l'ordre de leur apparition dans la spécification de l'application. Les libellés de ces champs sont justifiés à gauche.
- ☐ Les boutons de fonctions sont intégrés dans les fenêtres représentant les types d'entité et les types d'association, en bas de la dernière colonne.

Types de règles utilisées

- ☐ Internes, non modifiables car non visibles.

Forme des règles ergonomiques

- ☐ Règles imbriquées dans le code du programme.

II.2.5 UIDE

Nom général du système

UIDE (User Interface Design Environment)

Type général du système

User Interface Management system (UIMS)

Auteurs

Foley J. D., Kim W. C., Kovacevic S., Murray K., De Baar D., et Mullet K. E.,

Année de présentation

1992

Références bibliographiques

Foley J. D., Kim W. C., Kovacevic S. et Murray K., "*UIDE - An Intelligent User Interface Design Environment*", in J. Sullivan and S. Tyler (eds.), *Intelligent User Interfaces*, ACM Press, 1992, pp. 339-384

De Baar D., Foley J. D., et Mullet K. E., "*Coupling Application Design and User Interface Design*", in *Proceedings of the Conference on Human Factors in Computing Systems, CHI'92*, (Monterey, 3-7 mai 1992), pp. 259-266.

Liste et définition des objets de l'environnement

- *Data Model* :

C'est une base de données décrivant tous les objets de données utilisés par l'application, les relations entre ces objets, les actions sur les objets disponibles, les

préconditions et postconditions sur les actions, les attributs ou les propriétés des objets de données.

Le "data model" contient des métadonnées qui fournissent des précisions sur la manipulation des objets des données.

- Inference Engine :

Il utilise les actions et les attributs des objets de données pour gérer un ensemble d'objets interactifs concrets par manipulation directe.

Il comprend des règles de sélection d'OIC et des règles d'affichage qui prennent un ensemble d'objets d'interaction et une fenêtre mère comme base et produisent un arrangement spatial des OIC dans la fenêtre en résultat.

- Interface Design Environments :

- ☐ Open Look Developer's Guide (Devguide) : outil de conception d'interface et de spécification à manipulation directe.
- ☐ D2M2Edit : éditeur graphique à manipulation directe.

- Interaction Library :

Outil d'interface graphique (GUI toolkit) utilisé par l'Interface Design Environment pour produire une spécification textuelle.

- User Interface Description :

C'est une description textuelle fournie par l'Interface Design Environment (GIL) dans un format qui sert d'input au générateur de code de différents toolkits spécifiés pour créer des modules sources d'interface pour l'application. Il est possible d'ajouter des contrôles à des interfaces existants sous forme de modules sources et d'utiliser l'ensemble comme un input à l'outil de conception de l'affichage.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Le "data model", qui contient les objets de données.
- ☐ Les métadonnées utilisés par le concepteur pour clarifier les significations sémantiques des objets de données.
- ☐ Spécification textuelle et graphique.

Produits de la génération

- ☐ Menus
- ☐ Boîtes de dialogue complètes

Type et langage de spécification

- ☐ Spécification textuelle.

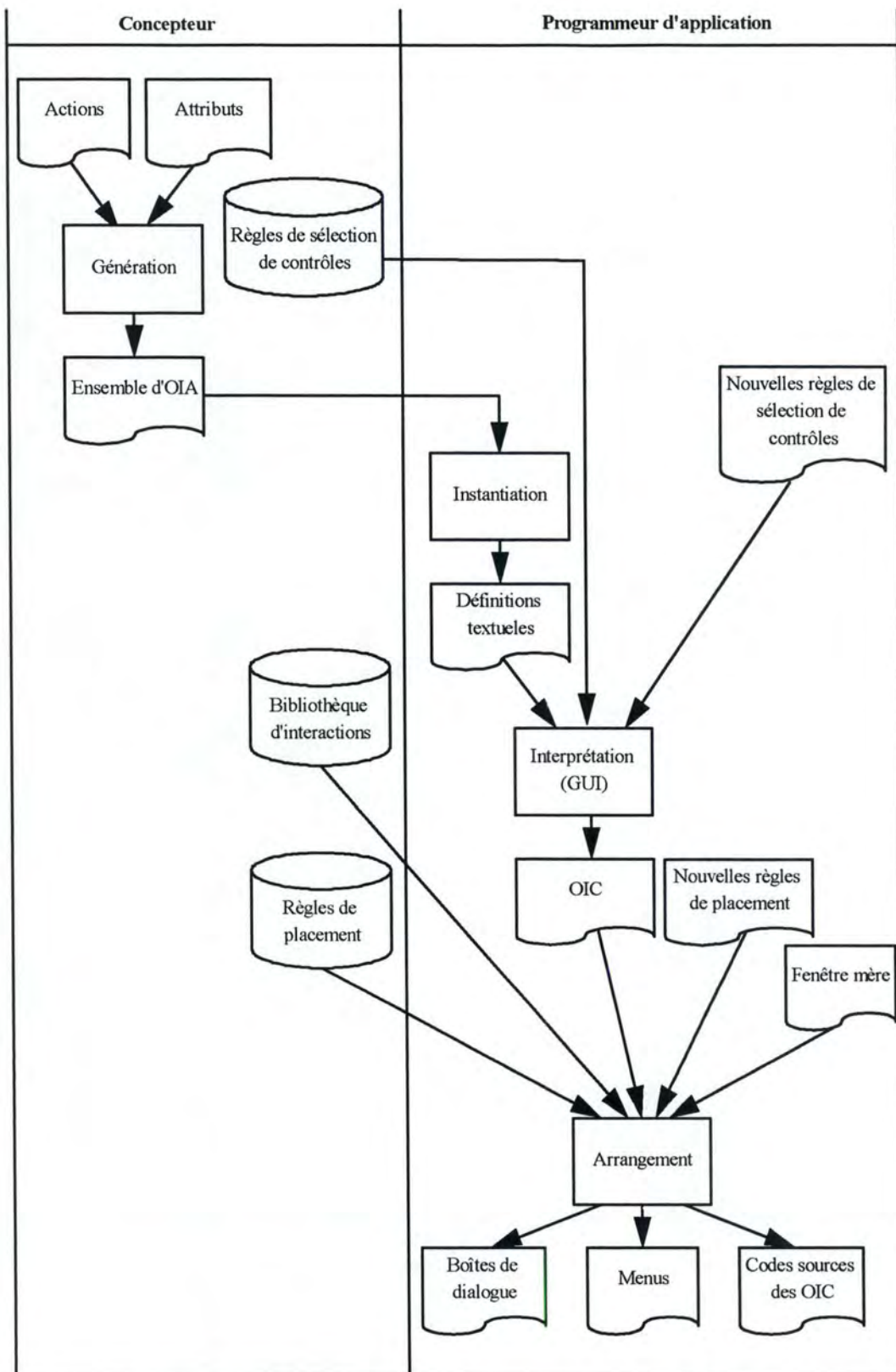


Figure 2.5 : Schéma de l'architecture générale de UIDE

Phase de sélection d'objets interactifs abstraits

Les règles de sélection de contrôle couvrent 5 types différents d'attributs : booléen, entier, réel, énumératif et chaîne de caractères.

Type booléen

On utilise des boîtes à cocher pour tout contrôle de type booléen.

Types entier et réel

La sélection est faite en partant de trois attributs de l'objet de données (= contenu, limite, étendu du domaine de valeurs) et d'un élément de métadonnées (= précision).

contenu	limites	étendu du domaine de valeurs	précision	widget
éditable	inconnu connu	quelconque petit large	quelconque basse haute basse haute	champ numérique échelle champ numérique échelle échelle avec champ de saisie
lecture uniquement	inconnu connu	quelconque quelconque	quelconque basse haute	champ de lecture de message jauge champ de lecture de message

Figure 2.6 : Sélection de contrôle pour type entier

contenu	limites	étendu du domaine de valeurs	précision	widget
éditable	inconnu connu	quelconque petit large	quelconque basse haute quelconque	champ de saisie de texte échelle champ de saisie de texte champ de saisie de texte
lecture uniquement	quelconque	quelconque	quelconque	champ de lecture de message

Figure 2.7 : Sélection de contrôle pour type réel

Type énumératif

Cette sélection est basée sur les caractéristiques suivantes :

- nombre total d'items,
- nombre d'items fixe ou variable,
- taille des libellés,
- nombre de choix minimal et maximal autorisés

Nbre total d'items	Nbre d'items variable/fixe	libellé	Minimum	Maximum	Widget
peu	statique	court	1	1	choix exclusif obligatoire
			0	1	choix exclusif facultatif
			0	≥ 1	choix non exclusif
		moyen	1	1	choix exclusif obligatoire
			0	1	choix exclusif facultatif
			0	≥ 1	boîte à cocher
		long	1	1	menu avec choix exclusifs obligatoires
			0	1	menu avec choix exclusifs facultatifs
			0	≥ 1	menu avec choix non exclusifs
	dynamique	quelconque	1	1	menu avec choix exclusifs obligatoires
			0	1	menu avec choix exclusifs facultatifs
			0	≥ 1	menu avec choix non exclusifs
beaucoup	quelconque	quelconque	1	1	liste de défilement abrégé
			0	1	liste de sélection simple
			0	≥ 1	liste de sélection multiple

Figure 2.8 : Sélection de contrôle pour type énumératif

Type chaîne de caractères

La sélection est basée sur le contenu et la longueur de la chaîne de caractères.

contenu	longueur	widget
éditable	courte	champ d'édition uni-linéaire
	longue	champ d'édition multi-linéaire
lecture uniquement	courte	champ uni-linéaire de lecture de messages
	longue	champ multi-linéaire de lecture de messages

Figure 2.9 : Sélection de contrôle pour type chaîne de caractères

Phase de placement d'objets interactifs concrets

La phase de placement d'objets interactifs concrets est présente. Dans ce placement, on peut distinguer :

- la localisation des OIC,
- l'arrangement spatial des contrôles à l'intérieur de la fenêtre de sortie,
- le dimensionnement des contrôles.

Une grille virtuelle d'affichage est utilisée pour aligner les objets et pour fournir une séparation visuelle adéquate et un regroupement des objets à l'intérieur de la fenêtre. La grille est découpée en unités dont la taille dépend de l'échelle de la fenêtre.

- Une unité de grille verticale sépare les contrôles ;
- une unité de grille sépare les ":" et le début de chaque contrôle ;
- deux unités de grille séparent les groupes de contrôles ;
- deux unités de grille séparent le bord de la fenêtre et le début du libellé le plus long ;

- les boutons de commande sont centrés au bas de la fenêtre.

Les objets interactifs concrets sont générés dans l'ordre de spécification.

Les items de la barre de menu sont arrangés au haut de la fenêtre mère.

Les items de menu sont assignés aux menus standards (file, view, edit, properties). Ceux qui ne peuvent pas être assignés aux 4 menus standards sont groupés sous un autre menu libellé "other".

Types de règles utilisées

- ☐ visibles
- ☐ modifiables

Forme des règles ergonomiques

- ☐ Base de règles "si ... alors ..." construites à partir des tables ci-dessus.

II.2.6 HUMANOID

Nom général du système

Humanoid

Type général du système

User Interface Management System (UIMS)

Auteurs

Szekeley P., Luo P. et Neches R.

Année de présentation

1992

Références bibliographiques

Szekeley P., Luo P. et Neches R., "*Facilitating the Exploration of Interface Design Alternatives : The HUMANOID Model of Interface Design*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'92, (Monterey, 3-7 mai 1992), pp. 507-515.

Szekeley P., Luo P. et Neches R., "*Beyond Interface Builders : Model-Based Interface Tools*", in Proceedings of the Conference on Human Factors in Computing Systems INTERCHI'93, (Amsterdam, 24-29 avril 1993), pp. 383-390.

Liste et définition des objets de l'environnement

- Générateur d'Interface :

Il reçoit la description sémantique de l'application ; cette description sémantique est enrichie de quelques annotations.

Il produit une interface graphique ; cette interface graphique peut être raffinée si de nouvelles informations sont ajoutées à la description sémantique.

- Bibliothèque d'OIC :

Elle contient la définition des boutons-radio, des boîtes à cocher, des boutons de commande, des listes de sélection, des tables, des graphes, des arbres et des types de fenêtres.

Il est possible d'ajouter de nouveaux gabarits (nouvelles définitions) dans cette bibliothèque. Le concepteur peut construire des interfaces en raffinant les gabarits de la bibliothèque.

- Bibliothèque de Comportements :

Elle contient un ensemble d'actions pouvant être appelées lorsqu'une manière d'agir standard de l'utilisateur sur un objet interactif a été rencontrée.

- Bibliothèque d'Attributs :

Elle contient un ensemble de méthodes qui implémentent des caractéristiques de groupes de commandes communément utilisées. Exemple : "*only-one-active*", qui signifie que dans le groupe de commandes, une seule commande peut être activée à un moment donné.

- Bibliothèque d'Effets de Bord (de commandes et de groupes de commandes) :

Elle contient un ensemble d'actions exécutées automatiquement lorsque certaines actions sont déclenchées par l'utilisateur. Exemple : un objet nouvellement créé devient automatiquement sélectionné.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Sémantique de l'application (pour la pré-génération)
- ☐ Interaction graphique (pour les raffinements à manipulation directe)

Produits de la génération

- ☐ Barres de menus
- ☐ Menus déroulants
- ☐ Boîtes de dialogue
- ☐ Fenêtres standards et fenêtres personnalisées
- ☐ Tables
- ☐ Graphes

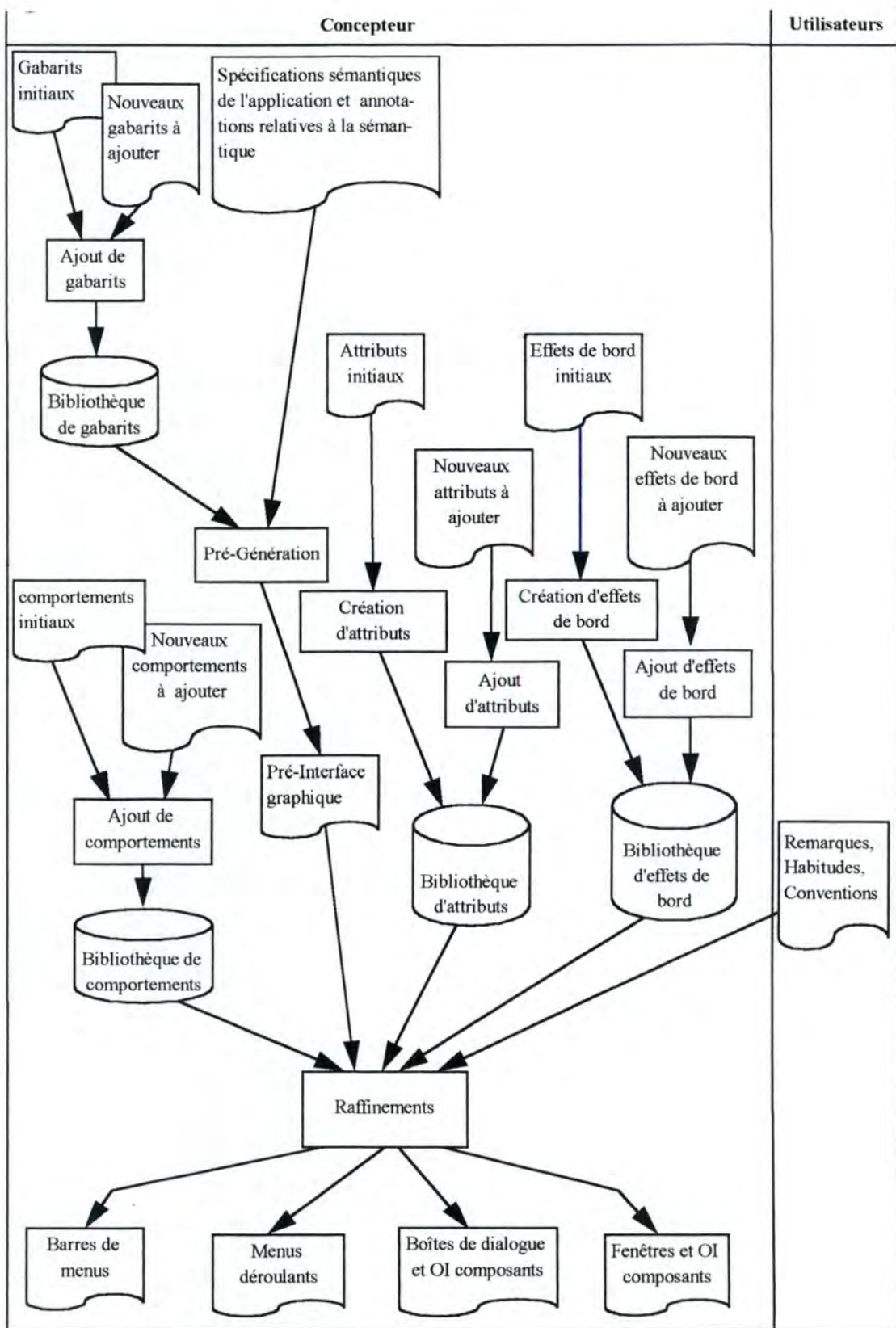


Figure 2.10 : Schéma de l'architecture générale de HUMANOID

Type et langage de spécification

Type et langage de spécification

- Langage déclaratif pour la pré-génération.
- Utilisation de formules de type feuilles de calcul pour spécifier des bouts de code (par exemple, pour définir des liens entre objets).
- Saisie graphique pour les raffinements.

La Sémantique de l'Application

Elle représente les objets et les opérations d'une application. Elle définit les domaines de sujet de l'interface. Elle est utilisée par le module de Présentation et le module des Comportements et pour le Séquencement du dialogue.

On utilise trois types d'objets pour spécifier la sémantique de l'application.

- les commandes :

la description d'une commande inclut la procédure qui implémente la commande, l'ensemble de ses préconditions et une description de chacune de ses données (input).

- les données :

la description d'une donnée inclut le type de valeur de la donnée, un prédicat qui est une procédure de validation de la donnée, les bornes inférieure et supérieure de la donnée, les *alternatives* qui spécifient un ensemble de valeurs possibles que la donnée peut prendre, un convertisseur de chaînes de caractères en valeurs de données et un convertisseur de valeurs de données en chaînes de caractères.

- les objets de l'application :

un objet de l'application regroupe un ensemble de commandes et d'objets ; par exemple, un programme de *mail* forme un objet de l'application.

La Présentation

Elle définit l'apparence visuelle des interfaces.

La bibliothèque des gabarits de Humanoid contient des gabarits pour :

- afficher les objets de l'application, les commandes et les objets de saisie de données,
- afficher des listes d'objets en colonne, en ligne ou dans des tables,
- afficher des graphes et des arbres,
- afficher plusieurs variantes de fenêtres "scrollables".

Le gabarit par défaut des objets de l'application peut générer une barre de menus déroulants, des panneaux de boutons de commandes, des panneaux de données globales utilisant des boutons-radio, des boîtes à cocher et d'autres techniques d'interaction traditionnelles.

Les Comportements

Ils définissent les actions qui peuvent être appliquées pour présenter les objets et leurs effets sur l'état de l'application et de l'interface ; par exemple, le clic de la souris.

Une spécification de comportement consiste en :

- une spécification de l'action qui invoque le comportement (exemples : le clic d'un bouton de souris, le déplacement de la souris),
- une spécification des parties de la présentation dans laquelle l'action s'applique (exemple : sur les widgets générés par un gabarit, sur toutes les parties d'un gabarit),
- une spécification des données de l'application sur lesquelles l'action s'applique et une spécification des "sous-actions" à prendre en compte durant les différentes phases de l'action (par exemple, pour le déplacement d'un objet à l'aide d'une souris, les points (phases) importants sont : pression du bouton de la souris, déplacement de la souris et relâchement du bouton de la souris).

Le Séquencement du Dialogue

Il définit les contraintes de séquencement pour l'exécution des commandes et l'octroi des données aux commandes.

La définition du séquencement inclut la spécification de l'ordre dans lequel les différents affichages apparaîtront à l'écran et l'ensemble des comportements disponibles à un moment donné. Humanoid détermine l'ensemble des comportements disponibles à un moment donné en se basant sur les contraintes de flux de données dans la sémantique de l'application et en appliquant un ensemble de règles prédéfinies à un modèle d'états de commandes (ou groupes de commandes) et aux données.

Exemples d'états de commandes :

- inactif (activable mais non sélectionné),
actif (sélectionné mais en attente de données nécessaires à son exécution),
en exécution ;
- disponible, non disponible ;
- prêt, pas prêt (une commande est prête quand elle est activée, toutes ses données sont correctes et aucune de ses préconditions n'est violée).

L'Action d'Effets de Bord

Elle définit les actions exécutées automatiquement quand des commandes ou des données de commande changent d'états (par exemple, un objet nouvellement créé est automatiquement sélectionné, fermer une boîte de dialogue peut réinitialiser toutes les options à leurs valeurs par défaut).

Le concepteur peut spécifier les effets de bord en écrivant des méthodes pour des transitions d'états appropriées ou en utilisant des attributs semblables aux attributs de séquencement. Par exemple :

- pour qu'une boîte de dialogue apparaisse en réponse à la sélection d'un menu, le concepteur peut écrire une méthode pour sa transition d'état "inactif-à-actif" ;
- pour qu'une donnée incorrecte introduite soit automatiquement remplacée par la valeur correcte précédente, le concepteur utilise l'attribut "*Revert-When-Incorrect*". Il existe d'autres attributs tels que "*Message-When-Incorrect*", "*Beep-When-Incorrect*".

Phase de sélection d'objets interactifs abstraits

- ☐ Absente
- ☐ La sélection d'objets interactifs concrets est réalisée explicitement par l'utilisateur.

Phase de placement d'objets interactifs concrets

- ☐ Présente.
- ☐ Placement par défaut des OIC, prédéterminé et unique, mais non décrit dans les articles.
- ☐ Lors du raffinement, en manipulation directe, l'utilisateur peut :
 - spécifier un gabarit, sélectionner le modèle de présentation ou une portion de gabarit et utiliser une boîte de dialogue (le Layout Editor) pour définir son placement. Cette boîte de dialogue permet de référencer l'objet et de définir son emplacement. L'utilisateur a le choix d'entrer une formule quelconque de type tableur pour définir le placement de l'objet relativement à d'autres objets.
 - utiliser la souris pour déplacer un objet, le dimensionner et le placer dans la zone désirée.

Types de règles utilisées

- ☐ Règles modifiables

Forme des règles ergonomiques

- ☐ Pas d'informations dans les articles.

II.2.7 DON

Nom général du système

DON

Type général du système

User Interface Presentation Design Assistant

Auteurs

Kim, W. C. et Foley, J. D.

Année de présentation

1990

Références bibliographiques

Kim, W. C. et Foley, J. D., "*DON : User Interface Presentation Design Assistant*", in Proceedings of the Symposium on User Interface Software and Technology, UIST'90, (Snowbird, octobre 1990), pp. 10-20.

Kim, W. C. et Foley, J. D., "*Providing High Level Control and Expert Assistance in the User Interface Presentation Design*", in Proceedings of INTERCHI'93, (Amsterdam, 24-29 avril '93), ACM Press, New York, 1993, pp. 430-437

Kim, W. C., "*Knowledge-Based Framework for an Automated User Interface Presentation Design Tool*", Ph. D. Dissertation, George Washington University, 14 février 1993.

Liste et définition des objets de l'environnement

- Organization Manager :

C'est un ensemble de règles qui analyse la spécification conceptuelle de haut niveau et le profil de préférence de haut niveau pour déterminer la structure organisationnelle. Il organise l'information qui va dans les menus et les boîtes de dialogue, fait la sélection d'objets d'interface appropriés et génère des structures logiques permettant d'organiser les objets. On y trouve les règles de structuration des menus et les règles d'organisation des boîtes de dialogue classées par ordre de priorité.

- Design Knowledge (base de connaissance de conception) :

Elle définit les relations de structure entre les instances d'objets de l'interface. Elle fournit également une charpente dans laquelle une description générée peut être traduite dans différents outils graphiques.

- Presentation Manager (Gestionnaire de Présentation) :

C'est un ensemble de règles qui prend en charge le placement des menus et des boîtes de dialogue. Ce placement est conforme à la base de connaissance de conception et aux principes de conception graphique.

Il vérifie que les décisions concernant le contenu informationnel et les types d'objets de l'interface ont été prises. Il est responsable de l'organisation et du placement des OIC dans les groupes logiques séparés par des séparateurs et des boîtes de regroupement.

On y trouve des règles de présentation de menus et des règles de présentation de boîtes de dialogue.

- Conceptual Design Knowledge (Base de Connaissance relative au Modèle Conceptuel) :

Cette base de connaissance contient la hiérarchie des classes des objets de l'application, les attributs des objets, les actions sur les objets de l'application, les unités d'information nécessaires pour les actions, les préconditions et les postconditions des actions. Son objectif est d'encapsuler la connaissance qui est essentielle pour la présentation des interfaces à l'utilisateur.

Les actions sont spécifiées à l'aide du langage IDL (Interface Definition Language).

- Preference Profile (Profil de Préférences) :

DON supporte le processus de spécification des préférences et maintient celles-ci dans le Profil de Préférences.

C'est une base à partir de laquelle le concepteur peut construire plusieurs profils de styles appropriés pour des niveaux particuliers des caractéristiques de l'utilisateur, des limitations du système et des outils graphiques utilisés.

Dans le Profil de Préférences, le concepteur peut modifier les priorités des critères structurant les objets qui, en fait, contrôlent les règles d'organisation par lesquelles les menus et les boîtes de dialogue sont structurés.

- *Style Knowledge* (Base de Connaissances de Styles) :

La Base de Styles contient les classes génériques d'objets d'interface. Elle augmente le contrôle individuel (du concepteur) du style de présentation des objets d'interface. Elle établit les styles par défaut des classes d'objets de l'interface. Elle est réutilisable pour de nouvelles applications.

Les objectifs visés par la Base de Styles sont :

- fournir les valeurs par défaut des attributs des objets interactifs ;
- permettre au concepteur de modifier les valeurs par défaut pour générer des styles personnalisés ;
- permettre au concepteur de spécialiser les objets de style pour des usages particuliers ;
- permettre au concepteur de regrouper un ensemble de valeurs d'attribut de style.

- *Graphical Design Knowledge* (Base de Connaissance de Conception Graphique) :

Les principes de conception graphique qui sont importants pour la conception d'interfaces utilisateur sont identifiés et classés par catégories telles que :

- connaissance des relations de localisation et de cohérence ayant trait aux objets d'interface ;
- domaines des valeurs d'attributs des objets d'interface ;
- contraintes sur les valeurs d'attributs des objets d'interface.

Les principes de conception graphique sont appliqués pendant la présentation des interfaces utilisateur pour améliorer la clarté visuelle, la cohérence et l'esthétique. Certains de ces principes sont :

- la proximité,
- la similitude,
- la direction,
- la séparation spatiale,
- l'équilibre dans la répartition des objets,
- la proportion
- le placement suivant une grille prédéfinie.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Description textuelle de la sémantique de l'application.
- ☐ Graphique appartenant à UIDE.

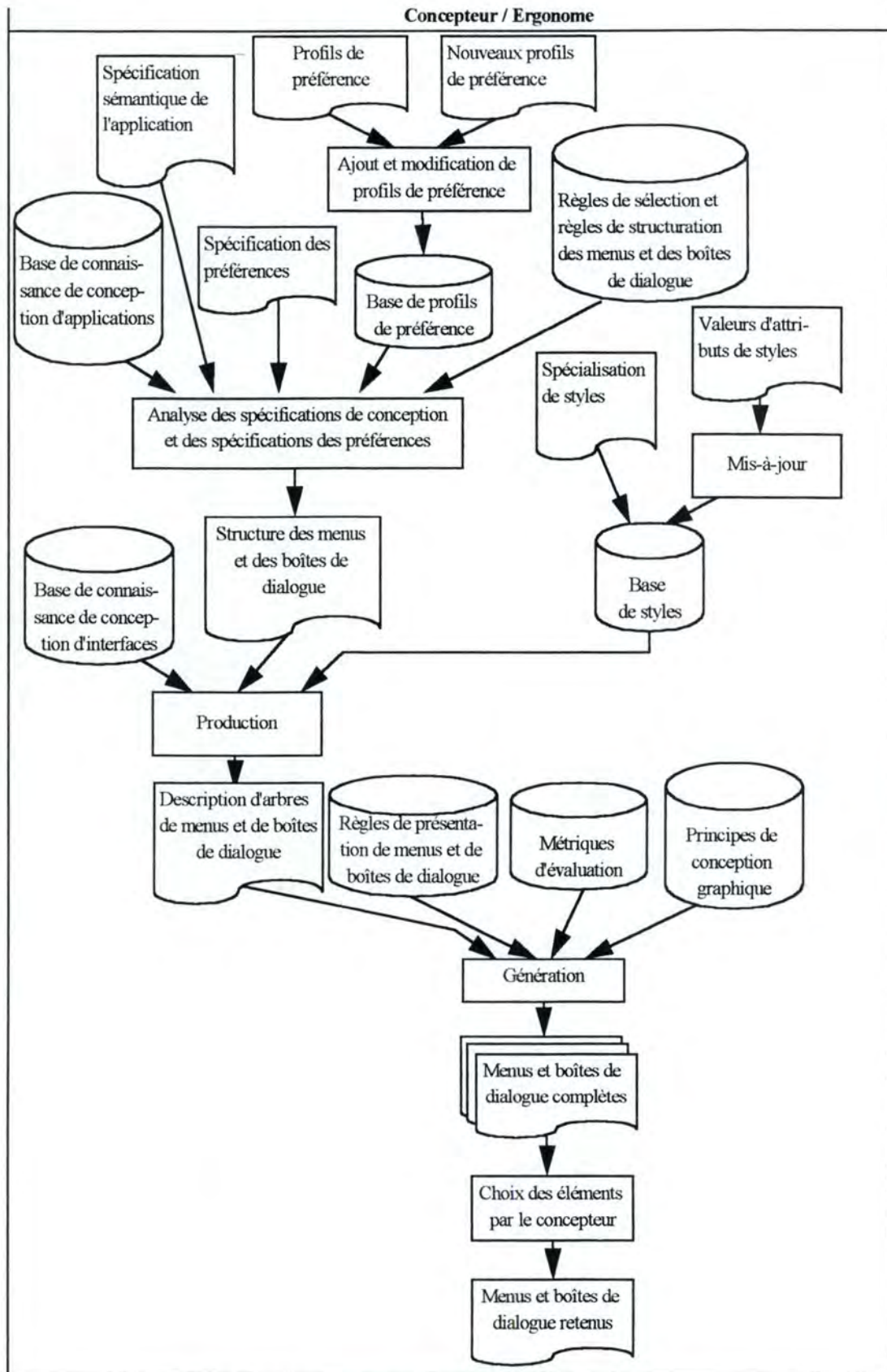


Figure 2.11 : Schéma de l'architecture générale de DON

Produits de la génération

- ☐ Menus,
- ☐ Boîtes de dialogue complètes.

Type et langage de spécification

- ☐ IDL

Phase de sélection d'objets interactifs abstraits

- ☐ Présente.
- ☐ Les commandes sont regroupées dans une structure d'arbre représentant un ensemble d'items dans un menu ou une boîte de dialogue répondant aux critères de spécification du concepteur et suivant l'ordre de priorité suivant :
 - 1- commandes opérant sur les mêmes objets ;
 - 2- classification des fonctions sémantiques de l'application (toutes les actions qui créent des objets d'application peuvent être organisées dans un même menu) ;
 - 3- les commandes qui opèrent sur les parties d'un même objet ;
 - 4- les commandes liées à une même tâche fonctionnelle ;
 - 5- les commandes ayant des paramètres communs ;
 - 6- les commandes ayant des préconditions et des postconditions similaires ;
 - 7- préconditions et postconditions séquentiellement dépendantes dans le temps (exemple : si la précondition d'une action B est satisfaite par la postcondition d'une action A, ces deux actions peuvent être items dans une même boîte de dialogue) ;
 - 8- ordonnancement par ordre d'importance ;
 - 9- classification par fréquence d'utilisation.
- ☐ Les attributs de style d'objets de menus, tels que l'ajustement des items, l'existence de titre d'une liste d'items, l'existence d'accélérateur et sa position, sont hérités des classes d'objets appropriées dans la Base de Connaissance de Style. De plus, les caractéristiques de menus, tels que la nécessité d'un choix d'item dans le menu, l'exclusion mutuelle des items, la compatibilité mutuelle des items, sont dérivées de la spécification des types d'attributs et sont représentées dans la Base de Connaissance de Conception (design knowledge).
- ☐ Les règles de sélection pour les boîtes de dialogue déterminent les objets d'interaction appropriés, c'est-à-dire choix, cycle, bouton-radio, boîte à cocher, échelle, champ de saisie, cadran, interrupteur (commutateur), liste, bouton de commande, etc ... Cette sélection est basée sur une analyse de l'information à représenter. En plus de l'information sur les types de données, les relations logiques entre les items de choix activés (mutuellement exclusifs, compatibles, d'ordre croissant ou décroissant) sont utilisées.

Phase de placement d'objets interactifs concrets

- ☐ Présente.
- ☐ Le placement se fait suivant l'algorithme qui suit :

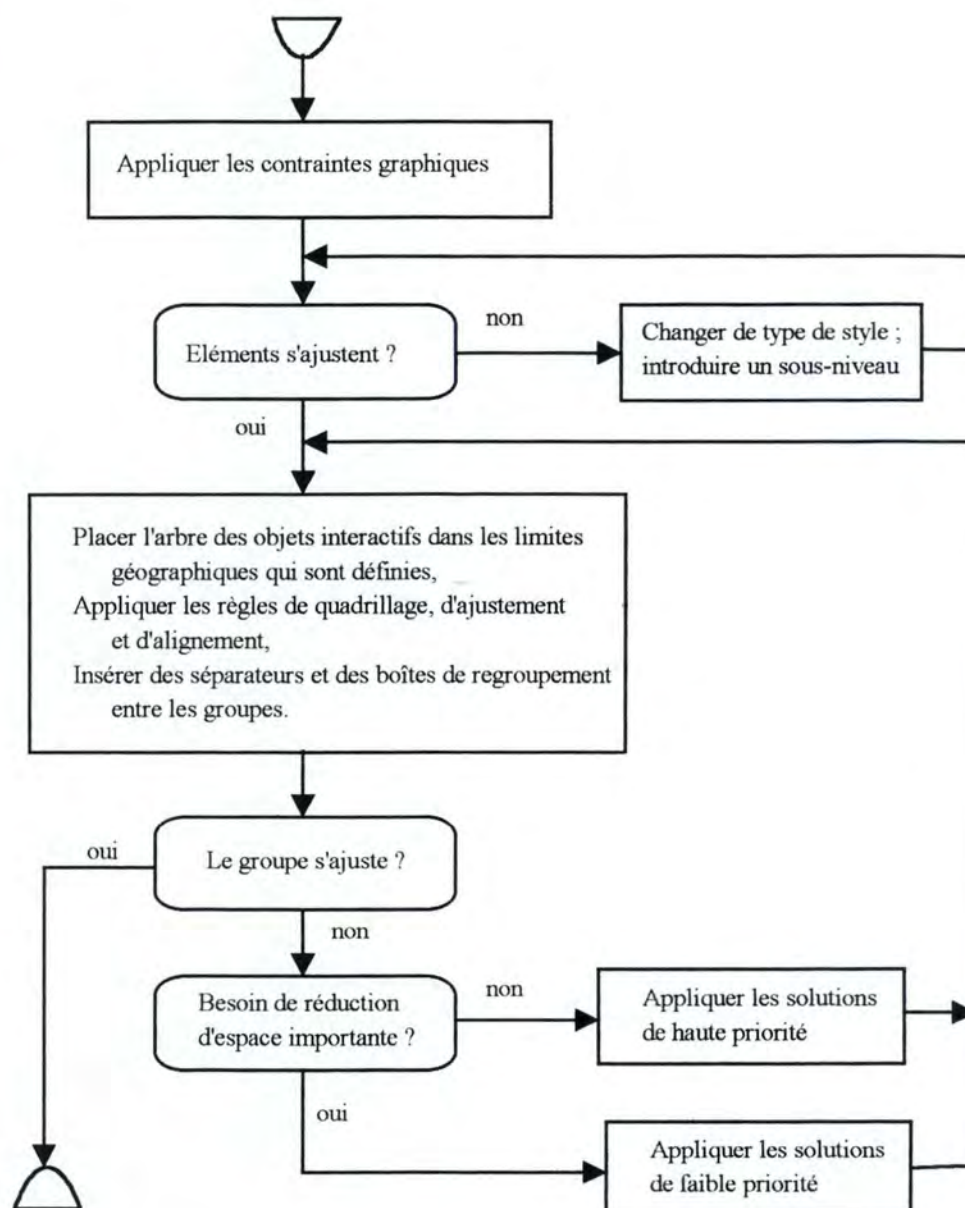


Figure 2.12 : Algorithme de placement de DON

Types de règles utilisées

- ☐ Non modifiables
- ☐ Visibles
- ☐ Explicites

Forme des règles ergonomiques

- ☐ Bases de connaissance
- ☐ Bases de règles
- ☐ Pour la sélection : base de règles sous forme "if ... then ... else..."
- ☐ Pour le placement : base de connaissance.

II.2.8 GENIUS

Nom général du système

GENIUS (GENerator for user Interfaces Using Software ergonomic rules)

Type général du système

Générateur d'interfaces.

Auteurs

Janssen C., Weisbecker A. et Ziegler J.

Année de présentation

1993

Référence bibliographique

Janssen C., Weisbecker A. et Ziegler J., "*Generating User Interfaces from Data Models and Dialogue Net Specification*", in Proceedings of the Conference on Human Factors in Computing Systems, INTERCHI'93, (Amsterdam, 24-29 avril 1993), pp. 418-423.

Liste et définition des objets de l'environnement

- L'Entity Relationship Diagram Editor (ERDE)

C'est un éditeur graphique à manipulation directe de modèles E/A. Il permet d'éditer les vues (ensemble d'attributs d'entités et d'associations) et de leur associer des spécifications textuelles qui ne peuvent être introduites graphiquement (stockées dans des *feuilles de propriété*). Ces spécifications textuelles contiennent :

- des propriétés descriptives comprenant le nom et une petite description. Pour un attribut, les propriétés sont, entre autres, le type, le domaine de valeur, le type de sélection et la valeur par défaut.
- des propriétés relatives à la tâche, comprenant la fréquence, le type d'accès et la priorité de la tâche.

Les fonctions associées aux vues peuvent également être décrites en tant que propriétés.

- Une *Base de Connaissance* :

Cette base de connaissance contient :

- des types d'objets interactifs abstraits,
- des règles de sélection et de placement des objets interactifs concrets.

Les types d'OIA contiennent des champs réservés aux données de l'application telles que le nom, le domaine, l'alignement, la couleur et la dimension.

Les règles sont inspirées des guides de conception existants, notamment :

- le guide OSF/MOTIF,
- le guide Open Look,
- le Common User Access d'IBM (1991),
- "Guidelines for Designing User Interface Software" de Smith, S. L. et Mosier, J.N. (1986).

- Un *User Interface Management System (UIMS)* :

OSF/Motif ou Open Look.

- Un *Système Expert* :

Ce système expert prend en amont le réseau de dialogue, les feuilles de propriété, les règles contenues dans les bases et fournit une description de l'interface utilisateur qui sera transformée en une spécification interprétable par le UIMS utilisé.

Schéma de l'architecture générale

Voir page suivante.

Source de génération

- ☐ Spécification graphique et textuelle des vues.
- ☐ Spécification textuelle des fonctions de manipulation de données.
- ☐ Spécification graphique et textuelle des fonctions de navigation.
- ☐ Spécification graphique du graphe de dialogue.

Produits de la génération

- ☐ Les objets interactifs concrets obtenus sont fonction de l'environnement de travail.
- ☐ Les OIC standards sont :
 - fenêtres,
 - boîtes de dialogue complètes.

Type et langage de spécification

- ☐ Saisie graphique.
- ☐ Description textuelle des attributs et des fonctions à l'aide de menus à sur-affichage.

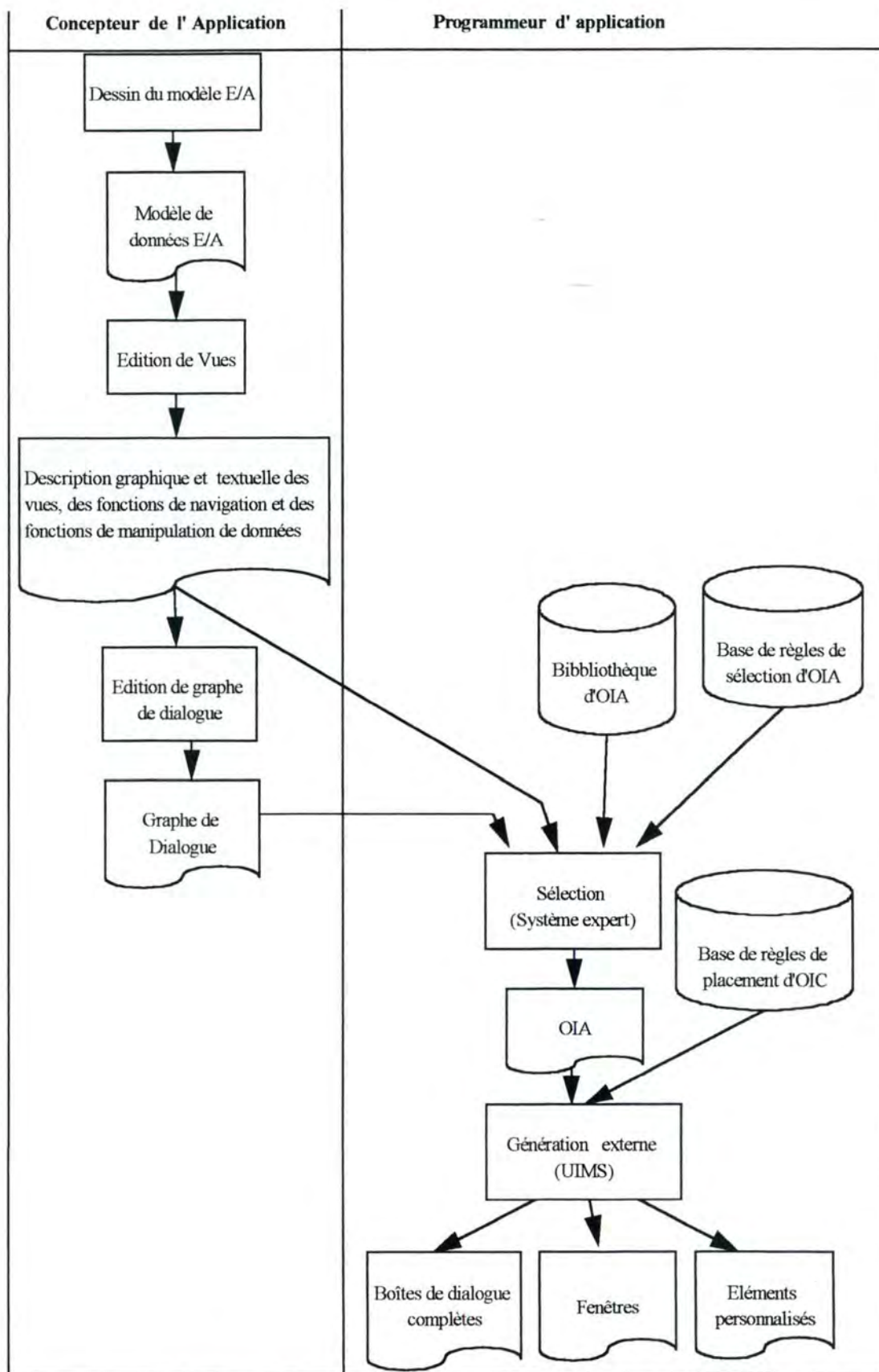


Figure 2.13 : Schéma de l'architecture générale de GENIUS

Phase de sélection d'objets interactifs abstraits

- ☐ Présente.
- ☐ Les OIA appropriés pour l'information contenue dans les vues sont sélectionnés en utilisant l'information associée aux vues. Pour chaque vue, une fenêtre est générée. Les OIA pour les données contenues dans la vue sont déterminés à partir des valeurs données dans les feuilles de propriété, principalement le type de donnée, le domaine, la condition de déclenchement. Par exemple, pour un champ de données avec des valeurs énumératives parmi lesquelles une seule sélection est permise à un moment donné, le système sélectionne une représentation en bouton-radio. Toutefois, si le nombre de valeurs possibles est élevé, un bouton-radio ne convient plus et le système choisira une liste de sélection.

Pour une fonction, son type, sa portée et sa fréquence d'utilisation déterminent sa représentation en tant qu'item de menu ou bouton de commande.

Phase de placement d'objets interactifs concrets

- ☐ Présente.
- ☐ Les règles sur les formats, l'arrangement et le placement sont tirées des guides de conception d'interface.
- ☐ L'arrangement à l'intérieur des éléments complexes tels que les groupes est d'abord effectué. Ensuite, les autres éléments sont disposés en fonction de l'espace disponible, de leur priorité ou de leur ordre de succession.

Types de règles utilisées

- ☐ Modifiables.

Forme des règles ergonomiques

- ☐ Base de règles modifiable.

II.3 CONCLUSION

A la fin de cette étude qui nous a conduits à faire ressortir les aspects qui nous semblent essentiels pour des outils d'aide à la génération automatique des interfaces utilisateur, il serait bon de tirer quelques conclusions générales.

Nous remarquerons tout d'abord que pour aider le concepteur et le programmeur dans leur tâche de conception d'interfaces utilisateur, plusieurs types d'outils sont proposés ou en voie de développement.

Les plus simples sont les boîtes à outils et les outils de conception d'interface et de spécification à manipulation directe. Ils permettent au programmeur de ne plus manipuler les points de l'écran pour obtenir ses interfaces utilisateur. Avec ces outils, la programmation des interfaces devient comparable à la programmation des autres modules de programme (tout en restant encore nettement plus consommatrice de temps). Ils sont aujourd'hui considérés comme des outils d'assez bas niveau. Toutefois, ils sont souvent utilisés par des outils de plus haut niveau. C'est surtout à ce titre qu'ils ont été rencontrés dans notre étude.

Viennent ensuite des outils plus évolués : générateur de vues, générateur de dialogue graphique, générateur d'interface, système de gestion d'interface utilisateur (UIMS), etc. Leur ambition est nettement plus grande : puisque l'essentiel de la sémantique des données échangées entre la machine et l'utilisateur peut se trouver (également) hors des modules de gestion d'interfaces, ne demander au concepteur que l'information supplémentaire et générer automatiquement les interfaces ; de plus, l'information supplémentaire sera spécifiée de la façon la plus simple possible, souvent à l'aide d'un langage graphique ou textuel déclaratif. Les outils que nous avons étudiés sont essentiellement de ce dernier type. Nous disons "essentiellement" parce que certains sont encore assez loin d'atteindre leur objectif ultime.

Les stratégies diffèrent d'un outil à un autre, mais on peut les résumer par les phases suivantes.

- Trouver la sémantique des données de l'application, en ayant recours à :
 - un éditeur de spécification souvent graphique et/ou textuel déclaratif,
 - au code du corps du programme, surtout à sa partie déclaration des données,
 - un langage de haut niveau qui permet d'ajouter des éléments de spécification au code du corps du programme ;
- Construire le schéma du dialogue machine-utilisateur, en particulier identifier les différentes phases du dialogue ;
- Trouver les OIA et leur hiérarchie ;
- Concrétiser les OIA en OIC ;
- Demander l'information manquante ; en particulier les libellés des OIC, les messages d'erreurs et les messages de guidance ;
- Résoudre le problème du placement :
 - problèmes d'"overflow",
 - localisation,
 - arrangement,
 - dimensionnement ;
- Raffinement par manipulation directe.

On ne retrouve pas nécessairement toutes ces phases dans un outil, et lorsqu'elles sont présentes, elles ne sont pas toujours traitées avec les mêmes moyens et la même profondeur. D'où la grande gamme d'outils.

Il est intéressant de remarquer que l'on retrouve dans la stratégie de la plupart d'outils une phase de sélection d'OIA. La sélection d'OIA avant conversion en OIC dans l'environnement cible permet une adaptation plus facile de l'outil à plusieurs environnements cibles.

Nous remarquerons également que, mis à part ITS qui laisse le placement des OIC sous la responsabilité du concepteur, tous les outils étudiés présentent une phase de placement d'OIC. Elle est plus ou moins élaborée suivant les outils, certains se contentant d'un dimensionnement et d'un arrangement minimaux et d'autres faisant d'elle un de leurs atouts essentiels. Nous constatons heureusement que la plupart d'outils permettent au concepteur de raffiner par manipulation directe les interfaces qu'ils génèrent de façon automatique ou semi-automatique.

Nous soulignerons enfin que dans la plupart des stratégies, les phases de sélection d'OIA, sélection d'OIC et placement d'OIC sont menés à bien en utilisant des bibliothèques d'objets (OIA, OIC, gabarits), des bases de règles (sélection et placement) et même des "bases de comportements". Dans la majeure partie des cas, ces bibliothèques et bases de règles et de comportements peuvent être modifiées de façon permanente ou exceptionnelle (uniquement pour une application donnée) par le concepteur ou l'expert en styles et de façon exceptionnelle par le programmeur. Ceci accroît incontestable la souplesse des outils concernés.

Dans le chapitre qui suit, nous reviendrons plus en détail sur DON, un outil qui nous semble très prometteur, surtout en ce qui concerne la façon dont il résout le problème du placement des OIC.

CHAPITRE III

LE PLACEMENT DANS DON

III.1 INTRODUCTION

A travers les différents outils étudiés, il s'avère important de consacrer une étude particulière à DON qui traite de façon remarquable les trois aspects principaux du placement : la localisation, le dimensionnement et l'arrangement.

Nous verrons l'algorithme DON et quelques aspects essentiels de la stratégie de placement. Nous appliquerons ensuite l'algorithme à un exemple.

Le programme offre deux méthodes de conception :

1°) Une méthode de conception itérative :

A partir de la spécification de l'application, le programme génère automatiquement une interface qui peut être raffinée itérativement par l'utilisateur.

2°) Une méthode de "génération et évaluation" :

Cette méthode génère plusieurs alternatives de conception et permet à l'utilisateur de visualiser l'évaluation métrique de chaque interface. Il peut ensuite agir itérativement sur l'interface de son choix afin de la raffiner.

III.2 ALGORITHME DE PLACEMENT

Nous nous proposons de donner les différentes étapes de l'algorithme de la méthode itérative. Les aspects essentiels de la stratégie seront développés dans le point III.3. Certaines étapes de cet algorithme sont aussi utilisées dans la deuxième méthode.

L'algorithme utilise un arbre de boîtes de dialogue qui est généré automatiquement en utilisant des règles de structuration du dialogue. Les feuilles de cet arbre sont les objets interactifs concrets qui apparaîtront dans l'interface (voir la structure de l'arbre figure 3.1). Les règles de placement s'appliquent en même temps à toutes les feuilles d'un même nœud mais pas aux feuilles de nœuds différents.

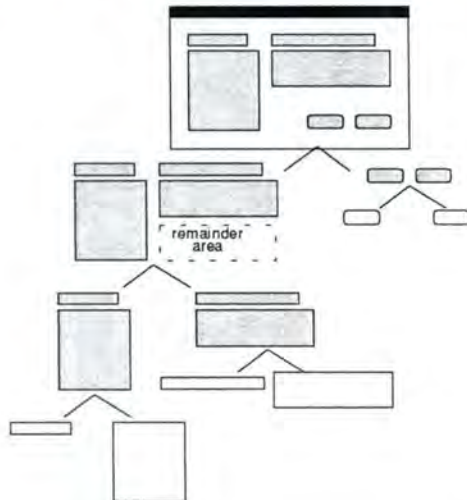


figure 3.1 : Exemple de placement utilisant la structure d'arbre de boîtes de dialogue.

Dans le but de générer automatiquement des interfaces attrayantes et esthétiques le placement dans DON prend essentiellement en compte la forme des objets interactifs concrets et la gestion de l'espace.

Les différentes phases de l'algorithme sont les suivantes :

- 1°) Recherche et traitement des situations d'overflow.
- 2°) Détermination de l'arrangement de la paire libellé-champ des objets interactifs concrets.
- 3°) Recherche de feuilles de formes similaires à l'intérieur de chaque groupe de nœud. Si un groupe comporte plusieurs feuilles de formes similaires, l'arbre est restructuré en mettant dans un même sous-groupe toutes ces feuilles de formes similaires.
- 4°) Traitement des feuilles d'un même groupe
 - (1) Toutes les feuilles d'un même nœud sont mises dans un pool de formes à placer ;
 - (2) Toutes les formes du pool sont analysées afin de pouvoir, si possible, les regrouper en se basant sur la similarité des formes ;
 - (3) Un critère de sélection est utilisé pour le choix de deux formes dans le pool.
L'algorithme utilise trois critères de sélection qui sont :
 - sélection de deux éléments en se basant sur la similarité de leurs formes et de leur tailles,
 - sélection de deux éléments en choisissant les plus gros,
 - sélection de deux éléments en choisissant les plus petits ;
 - (4) L'analyse des formes et des tailles est utilisée pour déterminer l'orientation et l'alignement des deux formes similaires (voir arbre de décision : figure 3.2) ;
 - (5) La paire ainsi formée est remise dans le pool des formes à placer comme un objet unique. L'information sur l'espace restant ("remainder area") est maintenue pour une utilisation éventuelle ;
 - (6) Les quatre derniers points sont réitérés jusqu'à ce que toutes les feuilles se réduisent à une seule forme.

Tout le traitement du 4°) se fait de façon récursive jusqu'à la détermination complète du placement de tous les nœuds.
- 5°) Traitement des boutons de confirmation OK et CANCEL.
- 6°) La dernière étape consiste en une évaluation métrique de l'interface finale (balance, symétrie, rapport des formes, etc.).

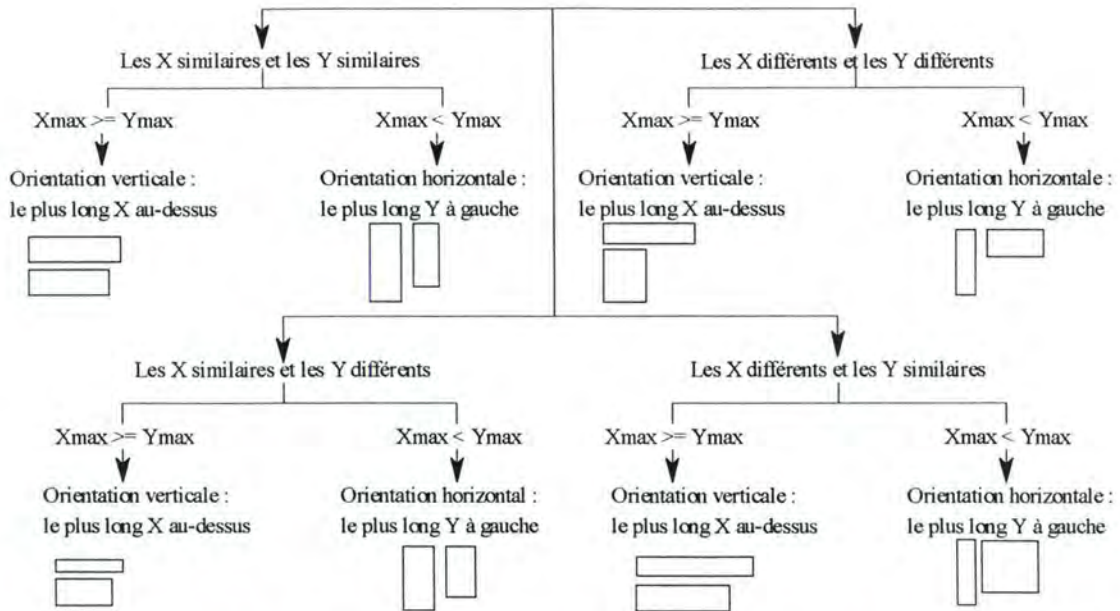


Figure 3.2 : Arbre de décision de la règle de placement pour une analyse de la forme et des dimensions de deux éléments rectangulaires.

La figure 3.3 montre quelques exemples de paires de formes similaires.

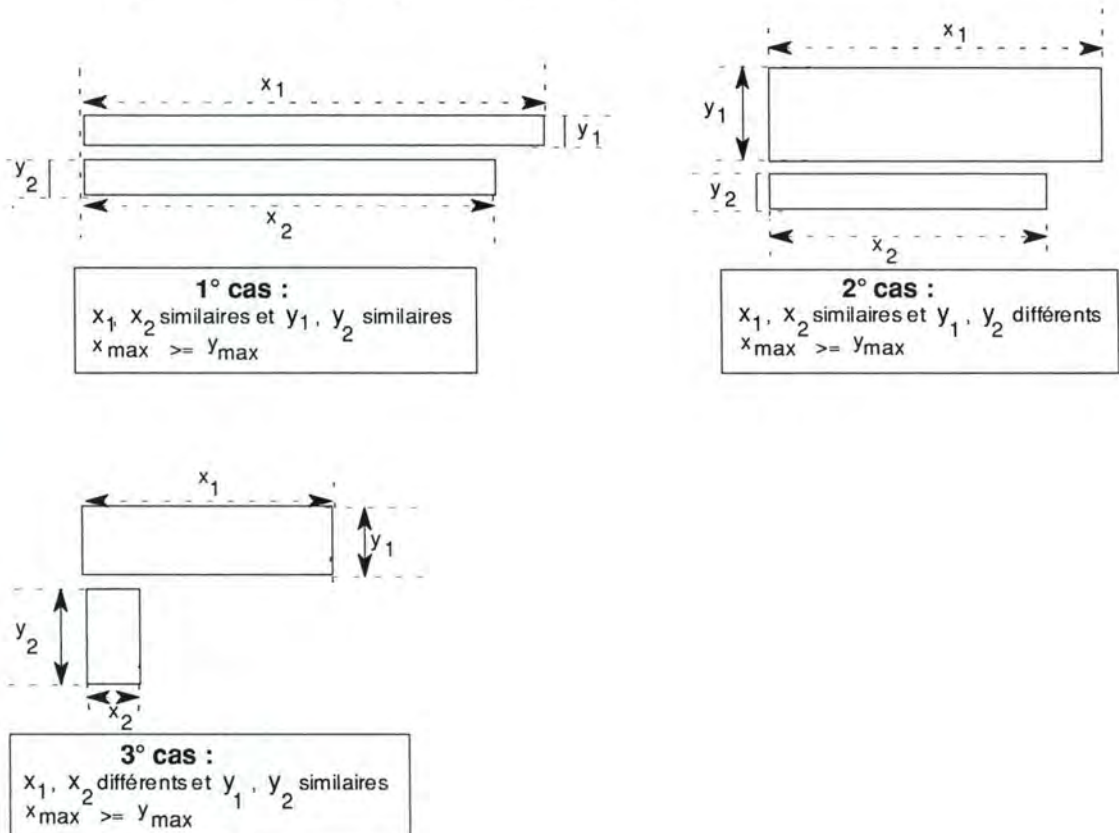


Figure 3.3 : Exemples de paires de formes similaires

III.3 QUELQUES ASPECTS ESSENTIELS DE LA STRATEGIE

III.3.1 ANALYSE DES FORMES ET DES DIMENSIONS

Une analyse de la forme et des dimensions est utilisée pour le placement du libellé et du champ d'affichage ou de saisie d'un item. Les deux éléments sont placés en recherchant la meilleure esthétique possible. Cette analyse permet aussi un regroupement des feuilles d'un nœud de formes *similaires*. Deux formes sont dites *similaires* si leurs dimensions sont *similaires*. On dit que deux dimensions x_1 et x_2 sont *similaires* si x_1 et x_2 sont telles que

$x_2 \geq x_1$ et $\frac{x_1}{x_2} \in] \frac{\sqrt{5}-1}{2}, 1]$. La valeur $\frac{\sqrt{5}-1}{2} = 0,618$ est donnée par défaut mais peut être changée par le concepteur de l'interface.

III.3.2 ANALYSE DE HAUT NIVEAU

Afin d'accroître l'efficacité de l'algorithme, une analyse de haut niveau est d'abord faite en examinant plusieurs formes à la fois. Elle permet d'identifier les cas simples où le nombre d'objets est petit, la possibilité de regroupements visuels basés sur la similarité des formes et des tailles, les situations "d'overflow". On dit qu'on a une situation "d'overflow" lorsque toutes les formes ne peuvent pas être placées dans la boîte de dialogue spécifiée pour insuffisance de place. L'analyse de haut niveau permet aussi d'estimer la conception "basée colonne" la mieux appropriée. DON traite d'abord ces différents cas avant d'appliquer l'algorithme de placement.

Traitement des cas simples

Lorsque le problème du placement se réduit au placement d'un message ou d'un OIC, cela se fait sans utiliser l'algorithme général. Sa résolution se limite à décider :

- 1°) de la localisation du libellé par rapport au champ,
- 2°) de la localisation et de l'orientation du groupe de boutons de configuration,
- 3°) de la taille de la boîte de dialogue.

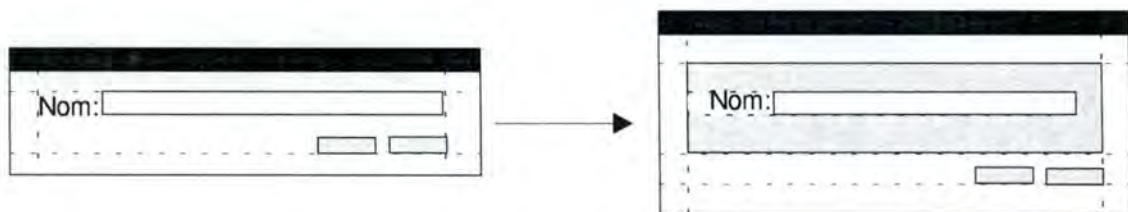


Figure 3.4 : Illustration de l'augmentation du contenu d'une forme afin d'obtenir la taille minimum requise pour une boîte de dialogue.

La solution "d'overflow"

Lorsque l'espace nécessaire aux éléments excède la taille de la boîte de dialogue, il faut par exemple :

- réduire les dimensions de certains objets d'interaction à leur minimum acceptable ;
- changer certains objets d'interaction par d'autres exigeant moins d'espace ;
- réduire les marges et les espacements.

Cas inverse de la situation "d'overflow"

Quand l'espace exigé par les objets d'interaction est moindre par rapport à la taille minimum de la boîte de dialogue, DON suggère un accroissement minimum de leur taille en ajoutant des blancs (voir figure 3.4).

Arrangement "basé colonne"

Si le nombre de formes horizontales est supérieur au nombre des formes non horizontales, alors il est suggéré un placement basé colonne (voir le 1^o cas de la figure 3.3).

Si la plupart des formes sont orientées verticalement alors un placement horizontal est recommandé (voir le 2^o cas de la figure 3.5).

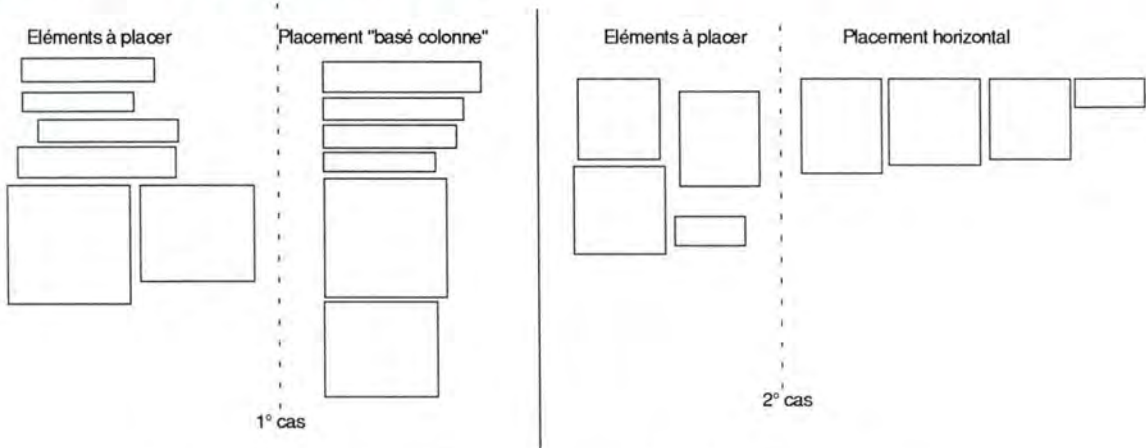


Figure 3.5 : Placement "basé colonne" et placement horizontal

III.3.3 CONTROLE DES MARGES ET DES ESPACEMENTS

L'algorithme de placement contrôle les marges et les espacements entre les items à l'intérieur des boîtes de dialogue. Il s'agit par exemple :

- de la marge de haut
- de la marge de bas
- des espaces entre les éléments des boîtes de dialogue
- de l'espacement entre les libellés et les champs d'un item

III.3.4 UTILISATION DE L'ESPACE RESTANT APPELE "REMAINDER AREA"

Le "remainder area" est l'espace libre inutilisé lorsque le placement d'un certain nombre d'OIC est déterminé (voir figure 3.6).

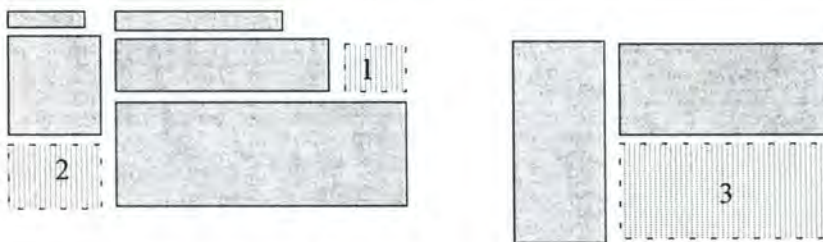


Figure 3.6 : Les surfaces hachurées 1, 2 et 3 représentent des exemples de "remainder area".

Pour l'utilisation de cet espace DON vérifie :

- 1°) Si un ou plusieurs éléments du pool (d'éléments à placer) peuvent entrer dans les limites du "remainder area", la règle utilisée est de prendre le plus gros dans le but de réduire au maximum cet espace vide. Ce processus est répété jusqu'à ce qu'aucun élément du pool ne puisse entrer dans les limites de l'espace inutilisé. Selon la figure 3.7, la forme A sera d'abord choisie puis la forme B.

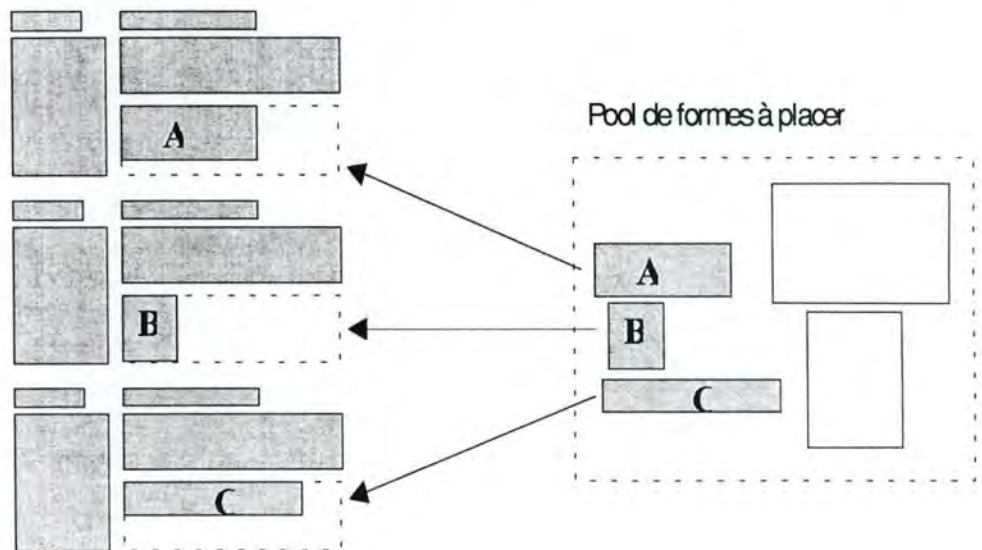


Figure 3.7 : Illustration de la stratégie d'alternatives de sélection d'un élément qui entre dans les limites du "remainder area" à partir du pool des formes à placer.

- 2°) Si aucun élément ne peut entrer dans les limites du "remainder area", DON vérifie s'il existe un ou plusieurs éléments qui satisfont aux conditions suivantes :
 - (1) chacun de ces éléments ne crée pas plus d'un "remainder area",
 - (2) le "remainder area" créé par le placement de chacun de ces éléments reste inférieur au "remainder area" précédent.

Il choisit parmi ces éléments celui qui crée le plus petit "remainder area" et le place sur l'espace inutilisé. La forme C de la figure 3.8 répond mieux à ces conditions.

S'il n'y a qu'un seul qui vérifie ces deux conditions, le choix se limite à cet élément. Si aucun élément ne vérifie les deux conditions le "remainder area" est fermé.

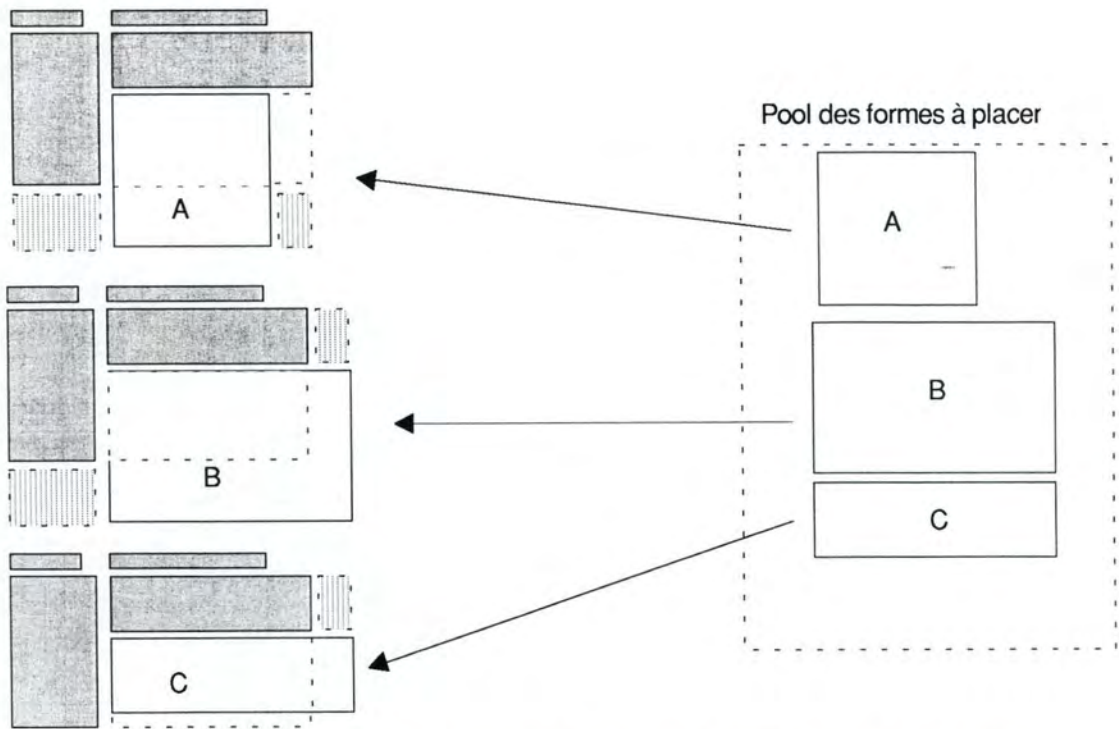


Figure 3.8 : Illustration de la stratégie d'alternatives de sélection d'un élément du pool des formes à placer lorsqu'aucun de ces éléments ne rentre dans les limites du "remainder area".

III.3.5 LES BOUTONS OK ET CANCEL

La localisation, l'alignement et l'orientation des boutons OK et CANCEL peuvent être explicitement spécifiés par le concepteur (voir figure 3.9).

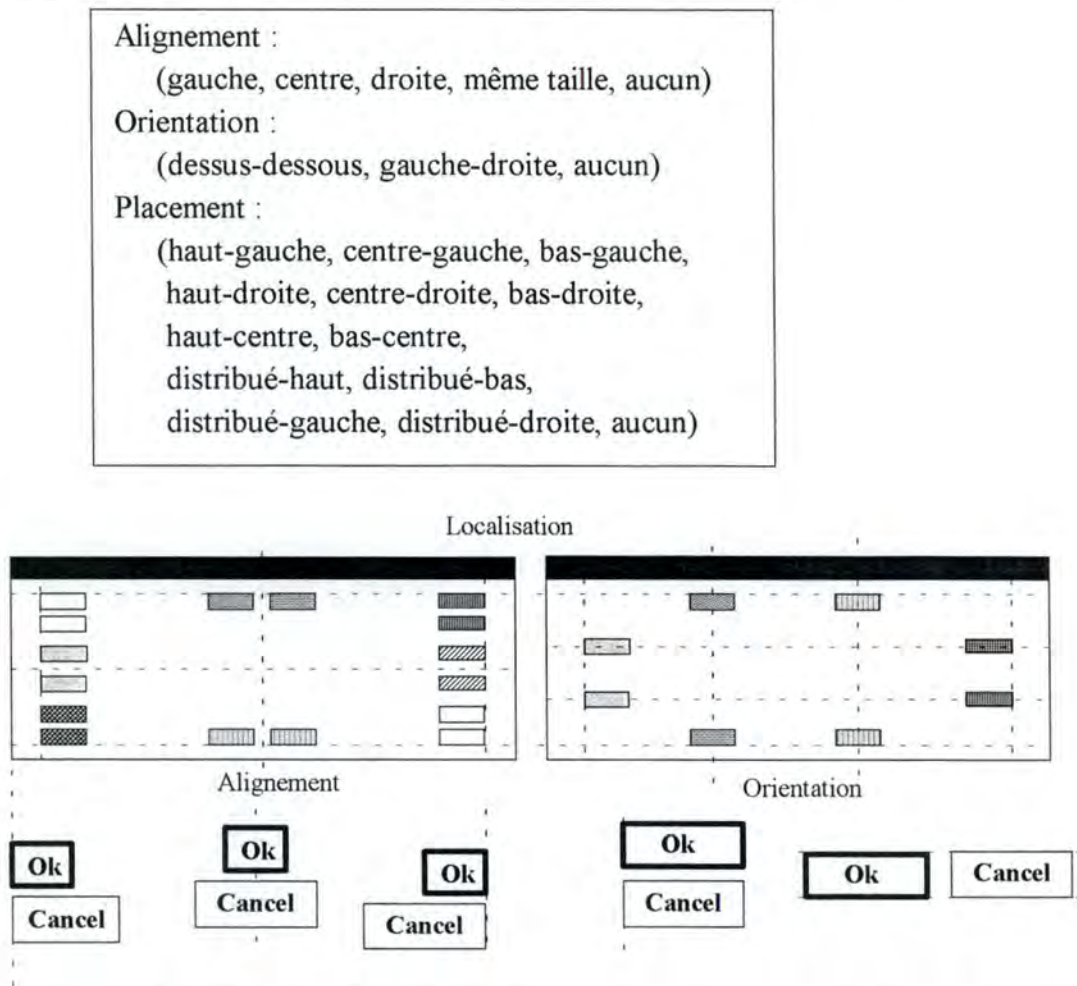


Figure 3.9 : Quelques exemples de localisation, d'alignement et d'orientation des boutons OK et Cancel.

Si elles ne sont pas spécifiées, les règles de placement tiennent compte de la taille et de la forme de l'espace occupé par les autres éléments ainsi que de l'espace disponible dans la détermination de la localisation et de l'orientation des boutons OK et CANCEL.

Deux cas peuvent se présenter :

- 1°) l'espace disponible est suffisant pour le placement des boutons OK et CANCEL qui sont considérés comme un tout. Si l'espace restant est horizontal, il est suggéré un placement horizontal des deux boutons. Si l'espace restant est vertical il est recommandé un placement vertical (voir figure 3.10).

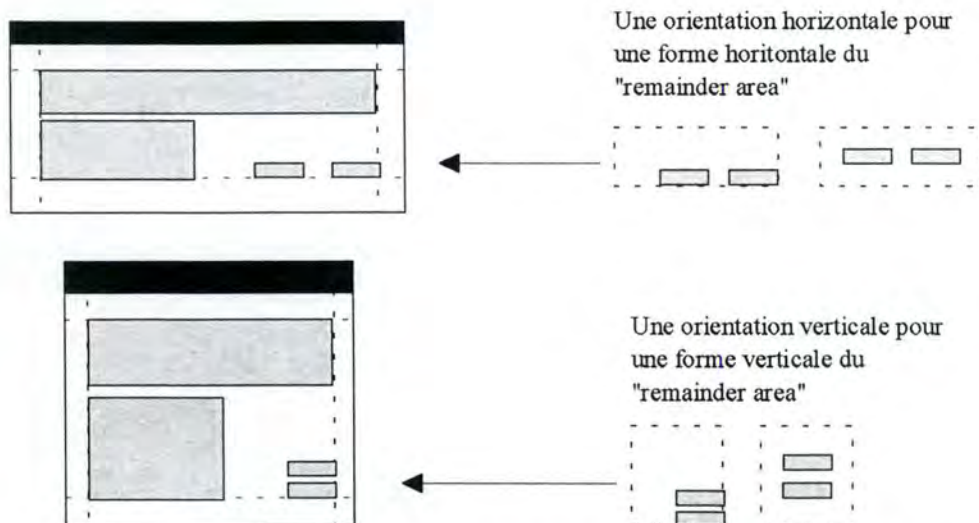


Figure 3.10 : Orientations par défaut des boutons de confirmation en se basant sur la forme du "remainder area".

2°) l'espace disponible est insuffisant. Dans ce cas, le "remainder area" est fermé et l'on obtient une forme horizontale ou verticale. Si elle est horizontale, il est suggéré un placement horizontal des deux boutons. Si elle est verticale, il est recommandé un placement vertical (voir figure 3.11).

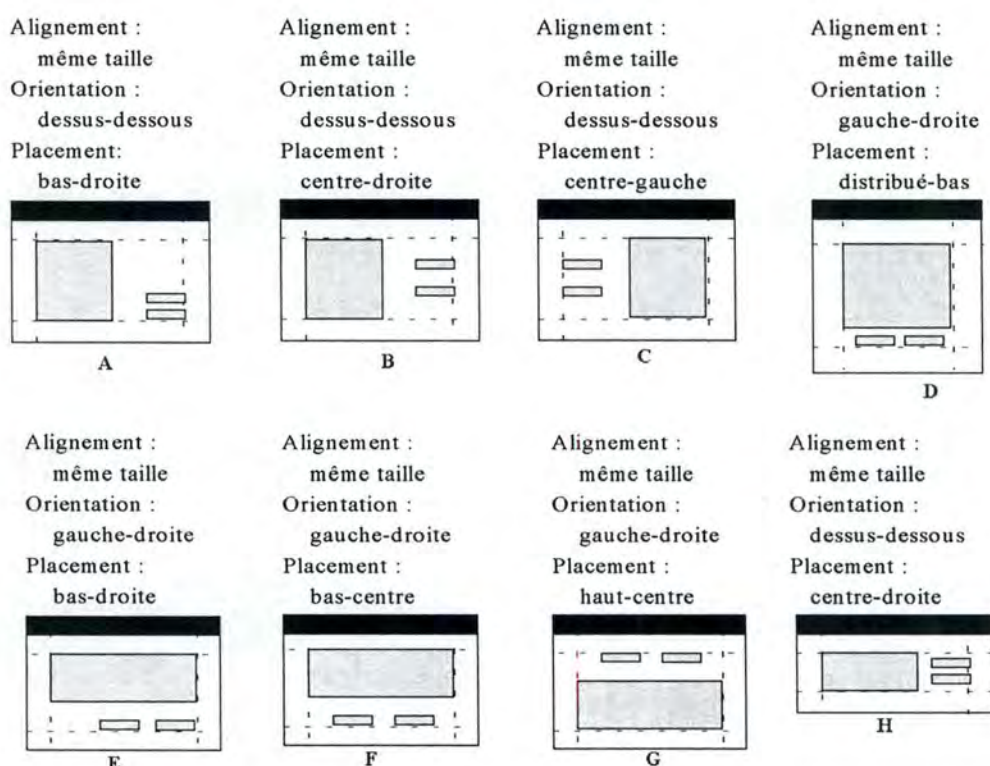
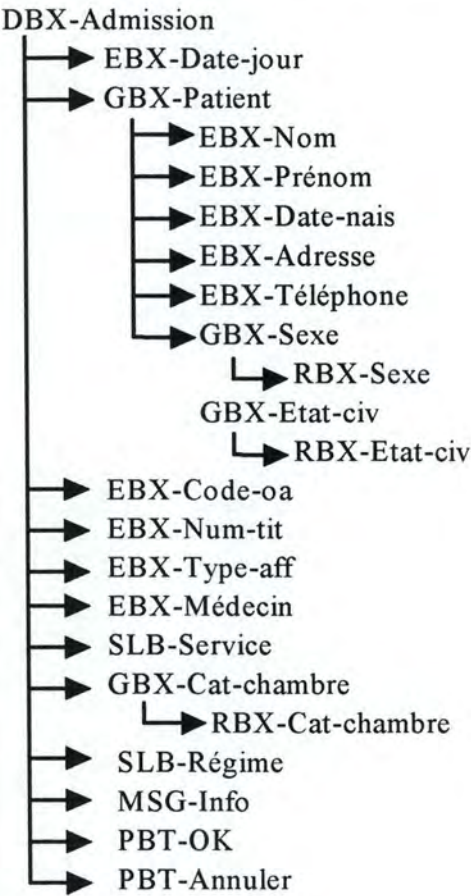


Figure 3.11 : Exemples de localisation et d'orientation des boutons de confirmation.

III.4 APPLICATION DE L'ALGORITHME A UN EXEMPLE

Nous illustrons l'algorithme de placement de DON en l'appliquant à un exemple. La boîte de dialogue choisie est supposée permettre la saisie des informations relatives à un patient lors de son hospitalisation.

Arbre de hiérarchie des noeuds :



Formes à placer :

Date du jour : (24)

Nom : (27)

Prénom : (30)

Date de naissance : (29)

Adresse : (41)

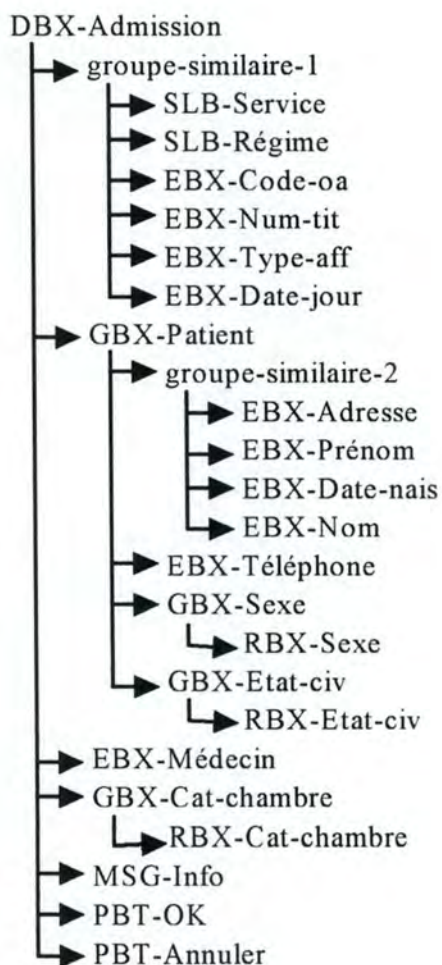
Phone : (21)

☒ masculin(13) ☐ féminin (12)

☐ célibataire (16) ☐ marié(10) ☒ veuf (9) ☐ divorcé (12)

Code de l'organisme :	(29)	
Numéro de titulaire :	(29)	
Type d'affiliation :	(26)	
Médecin :	(17)	
Service :	(31)	
<input type="checkbox"/> chambre particulière (25)	<input checked="" type="checkbox"/> chambre à 2 lits (21)	<input type="checkbox"/> chambre à 4 lits (21)
Régime :	(30)	
OK	Cancel	

Après l'analyse pour regroupement de feuilles de formes similaires, on obtient l'arbre de structuration suivant :



Arrangement de la boîte de regroupement Patient

Arrangement de groupe-similaire-2

Adresse :	(41)
Prénom :	(30)
Date de naissance :	(29)
Nom :	(27)

Arrangement de GBX-Sexe

☒ masculin(13)

☐ féminin (12)

→

Sexe

☒ masculin(13)

☐ féminin (12)

Arrangement de GBX-Etat-Civ

(Les items Célibataire, Divorcé et Marié sont similaires ; on les arrange d'abord avant de prendre en compte l'item veuf)

☐ célibataire (16)

☐ divorcé (12)

☐ marié(10)

→

☐ célibataire (16)

☐ divorcé (12)

☐ marié(10)

☒ veuf (9)

→

Etat Civil

☐ célibataire (16)

☐ divorcé (12)

☐ marié(10)

☒ veuf (9)

(GBX-Etat-Civ et GBX-Sexe sont similaires)

Etat Civil

☐ célibataire (16)

☐ divorcé (12)

☐ marié(10)

☒ veuf (9)

Sexe

☒ masculin(13)

☐ féminin (12)

(EBX-Téléphone et le groupe précédent ont des longueurs similaires)

Téléphone : (25)	
Etat Civil	
<input type="checkbox"/>	célibataire (16)
<input type="checkbox"/>	divorcé (12)
<input type="checkbox"/>	marié(10)
<input checked="" type="checkbox"/>	veuf (9)
Sexe	
<input checked="" type="checkbox"/>	masculin(13)
<input type="checkbox"/>	féminin (12)

Arrangement de GBX-Patient

Adresse : (41)	
Prénom : (30)	
Date de naissance : (29)	
Nom : (27)	
Téléphone : (25)	
Etat Civil	
<input type="checkbox"/>	célibataire (16)
<input type="checkbox"/>	divorcé (12)
<input type="checkbox"/>	marié(10)
<input checked="" type="checkbox"/>	veuf (9)
Sexe	
<input checked="" type="checkbox"/>	masculin(13)
<input type="checkbox"/>	féminin (12)

→

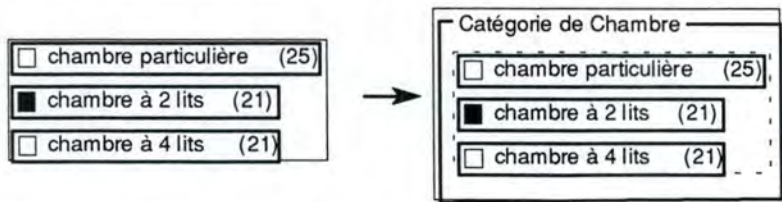
Patient	
Adresse : (41)	
Prénom : (30)	
Date de naissance : (29)	
Nom : (27)	
Téléphone : (25)	
Etat Civil	
<input type="checkbox"/>	célibataire (16)
<input type="checkbox"/>	divorcé (12)
<input type="checkbox"/>	marié(10)
<input checked="" type="checkbox"/>	veuf (9)
Sexe	
<input checked="" type="checkbox"/>	masculin(13)
<input type="checkbox"/>	féminin (12)

Arrangement de groupe-similaire-1

Service : (31)	
Régime : (30)	
Code de l'organisme : (29)	
Numéro de titulaire : (29)	
Type d'affiliation : (26)	
Date du jour : (24)	

Arrangement de GBX-Cat-Chambre

(Les trois items sont similaires)



Placement des groupes obtenus

(Groupe-similaire-1 et RBX-Cat-Chambre sont similaires)

The form shows the placement of groups. It includes fields for Service (31), Régime (30), Code de l'organisme (29), Numéro de titulaire (29), Type d'affiliation (26), Date du jour (24), and a 'Catégorie de Chambre' group containing 'chambre particulière (25)', 'chambre à 2 lits (21)', and 'chambre à 4 lits (21)'.

(GBX-Patient et le groupe précédent ont des hauteurs similaires ; il y a un "remainder area" appréciable)

The form shows the placement of GBX-Patient and the previous group. The Patient group includes fields for Adresse (41), Prénom (30), Date de naissance (29), Nom (27), Téléphone (25), Etat Civil (célibataire 16, divorcé 12, marié 10, veuf 9), and Sexe (masculin 13, féminin 12). The previous group includes Service (31), Régime (30), Code de l'organisme (29), Numéro de titulaire (29), Type d'affiliation (26), Date du jour (24), and a 'Catégorie de Chambre' group. A shaded 'remainder area' is shown below the previous group.

(Le "remainder area" est exploité pour le placement de EBX-Médecin)

Patient		Service : (31)	
Adresse : (41)		Régime : (30)	
Prénom : (30)		Code de l'organisme : (29)	
Date de naissance : (29)		Numéro de titulaire : (29)	
Nom : (27)		Type d'affiliation : (26)	
Téléphone : (25)		Date du jour : (24)	
Etat Civil		Catégorie de Chambre	
<input type="checkbox"/> célibataire (16)		<input type="checkbox"/> chambre particulière (25)	
<input type="checkbox"/> divorcé (12)		<input checked="" type="checkbox"/> chambre à 2 lits (21)	
<input type="checkbox"/> marié(10)		<input type="checkbox"/> chambre à 4 lits (21)	
<input checked="" type="checkbox"/> veuf (9)			
Sexe		Médecin : (17)	
<input checked="" type="checkbox"/> masculin(13)			
<input type="checkbox"/> féminin (12)			

Ajout de la zone de message

MSG-Info			
Patient		Service : (31)	
Adresse : (41)		Régime : (30)	
Prénom : (30)		Code de l'organisme : (29)	
Date de naissance : (29)		Numéro de titulaire : (29)	
Nom : (27)		Type d'affiliation : (26)	
Téléphone : (25)		Date du jour : (24)	
Etat Civil		Catégorie de Chambre	
<input type="checkbox"/> célibataire (16)		<input type="checkbox"/> chambre particulière (25)	
<input type="checkbox"/> divorcé (12)		<input checked="" type="checkbox"/> chambre à 2 lits (21)	
<input type="checkbox"/> marié(10)		<input type="checkbox"/> chambre à 4 lits (21)	
<input checked="" type="checkbox"/> veuf (9)			
Sexe		Médecin : (17)	
<input checked="" type="checkbox"/> masculin(13)			
<input type="checkbox"/> féminin (12)			

Ajout des boutons de confirmation dans le "remainder area"

MSG-Info	
Patient	
Adresse : (41)	Service : (31)
Prénom : (30)	Régime : (30)
Date de naissance : (29)	Code de l'organisme : (29)
Nom : (27)	Numéro de titulaire : (29)
Téléphone : (25)	Type d'affiliation : (26)
	Date du jour : (24)
Etat Civil	Catégorie de Chambre
<input type="checkbox"/> célibataire (16)	<input type="checkbox"/> chambre particulière (25)
<input type="checkbox"/> divorcé (12)	<input checked="" type="checkbox"/> chambre à 2 lits (21)
<input type="checkbox"/> marié(10)	<input type="checkbox"/> chambre à 4 lits (21)
<input checked="" type="checkbox"/> veuf (9)	
Sexe	Médecin : (17)
<input checked="" type="checkbox"/> masculin(13)	
<input type="checkbox"/> féminin (12)	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Ajout du titre et du cadre de l'interface

ADMISSION	
MSG-Info	
Patient	
Adresse : (41)	Service : (31)
Prénom : (30)	Régime : (30)
Date de naissance : (29)	Code de l'organisme : (29)
Nom : (27)	Numéro de titulaire : (29)
Téléphone : (25)	Type d'affiliation : (26)
	Date du jour : (24)
Etat Civil	Catégorie de Chambre
<input type="checkbox"/> célibataire (16)	<input type="checkbox"/> chambre particulière (25)
<input type="checkbox"/> divorcé (12)	<input checked="" type="checkbox"/> chambre à 2 lits (21)
<input type="checkbox"/> marié(10)	<input type="checkbox"/> chambre à 4 lits (21)
<input checked="" type="checkbox"/> veuf (9)	
Sexe	Médecin : (17)
<input checked="" type="checkbox"/> masculin(13)	
<input type="checkbox"/> féminin (12)	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Interface générée

ADMISSION

MSG-Info

Patient

Adresse : (41)

Prénom : (30)

Date de naissance : (29)

Nom : (27)

Téléphone : (25)

Etat Civil

☐ célibataire (16)

☐ divorcé (12)

☐ marié(10)

☒ veuf (9)

Sexe

☒ masculin(13)

☐ féminin (12)

Service : (31)

Régime : (30)

Code de l'organisme : (29)

Numéro de titulaire : (29)

Type d'affiliation : (26)

Date du jour : (24)

Catégorie de Chambre

☐ chambre particulière (25)

☒ chambre à 2 lits (21)

☐ chambre à 4 lits (21)

Médecin : (17)

OK

Cancel

III.5 CONCLUSION

Nous constatons à partir de cet exemple combien la stratégie de placement de DON est élaborée et permet d'obtenir des interfaces esthétiques. Elle minimise également les espaces libres non utilisés.

Nous remarquons cependant que l'algorithme ne respecte pas l'ordre logique de succession des OIC lorsque ceux-ci sont des feuilles d'un même nœud. Toutefois une alternative est offerte à l'utilisateur qui peut raffiner son interface par manipulation directe et résoudre ainsi le problème de la succession logique des OIC.

Nous reviendrons plus en détail sur certains atouts et inconvénients de la stratégie de placement de DON dans le paragraphe IV.4.

Dans le prochain chapitre nous traiterons du placement dans TRIDENT qui applique une autre stratégie.

CHAPITRE IV

PLACEMENT DES OBJETS INTERACTIFS CONCRETS DANS TRIDENT

IV.1 PRE-REQUIS

Dans cette partie, nous proposons un aperçu partiel des différentes étapes de la démarche de conception des interfaces dans TRIDENT telle qu'elle se présente pour notre travail. Cette démarche est schématisée à la figure 1 où EX représente un exemple d'application.

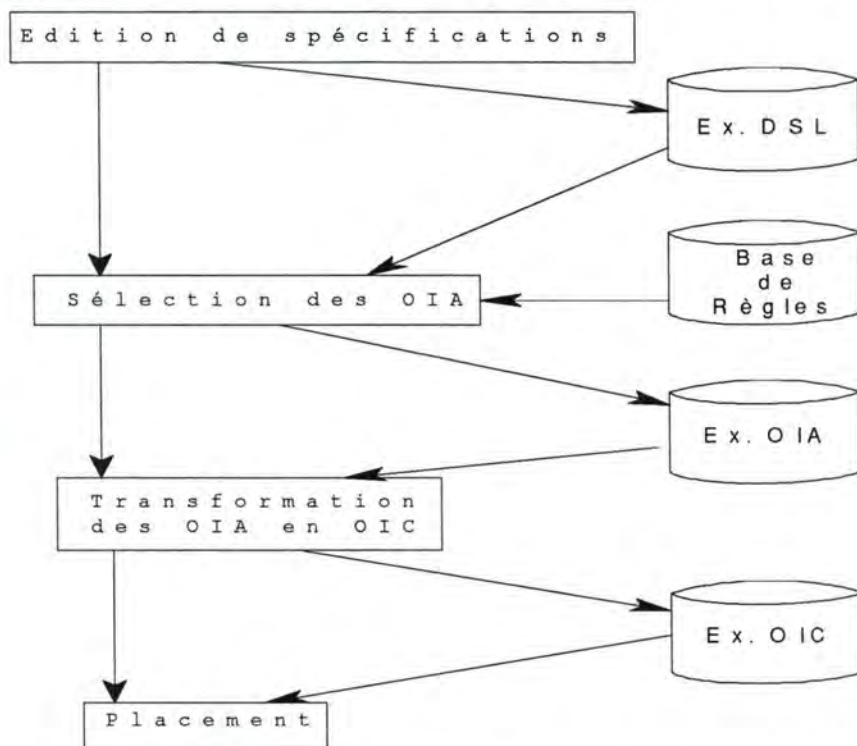


figure 1 : Démarche de conception des interfaces dans TRIDENT

- 1°) A partir d'un *Editeur* de spécification (PAQUITA), le concepteur spécifie soit graphiquement soit textuellement la sémantique de l'application à l'aide de *DSL* (Dynamic Specification Language), un langage de spécification déclaratif de haut niveau qui tient compte des aspects dynamiques. Ce langage est en cours d'extension en vue d'inclure les aspects ergonomiques.
- 2°) Les spécifications *DSL* sont enregistrées dans deux bases de données textuelles :
 - a) La première base de données contient les spécifications traditionnelles de la structure des données de l'application notamment sous forme d'un schéma Entité-Association.
 - b) La deuxième base de données contient un schéma du modèle du dialogue (ici, Ex. DSL).
- 3°) A l'aide de règles contenues dans une base de règles, le *sélecteur d'OIA* sélectionne automatiquement les OIA à partir des spécifications et crée des *spécifications d'OIA* qui sont stockées dans une base de données (ici, Ex OIA). Ces spécifications tiennent compte de la hiérarchie des OIA qui est représentée sous forme d'arbre (voir figure 2).

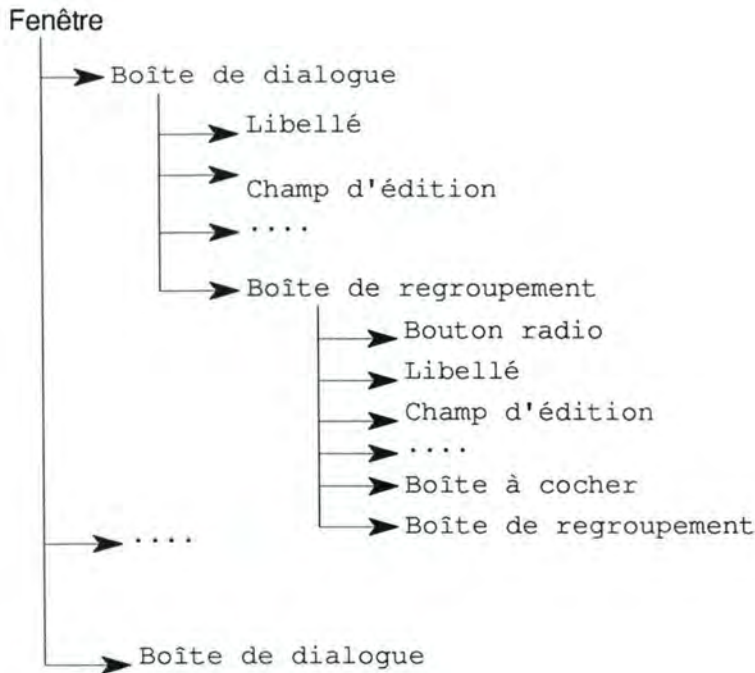


Figure 2 : Arbre de hiérarchie des OIA

- 4°) Etant donné que les spécifications des OIA sont indépendantes de l'environnement, les OIA sont transformés en OIC pour tenir compte d'un environnement cible choisi. Ces OIC sont stockés dans une base de données (ici, Ex. OIC).
- 5°) A partir de cette dernière base de données, les OIC peuvent être générés physiquement et placés automatiquement dans l'interface suivant les trois aspects : localisation physique, dimensionnement et arrangement de l'objet suivant des règles ergonomiques appropriées.

Nous nous intéresserons à cette dernière partie consacrée au placement des objets interactifs concrets.

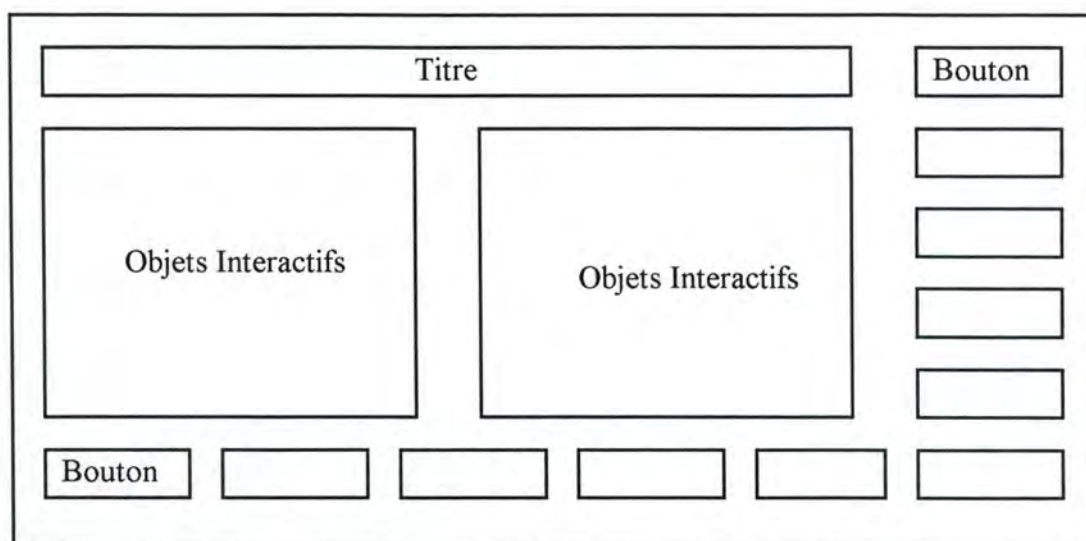
IV.2 ALGORITHME DE PLACEMENT

IV.2.1 PLACEMENT DES OIC POUR UNE BOITE DE DIALOGUE MODALE

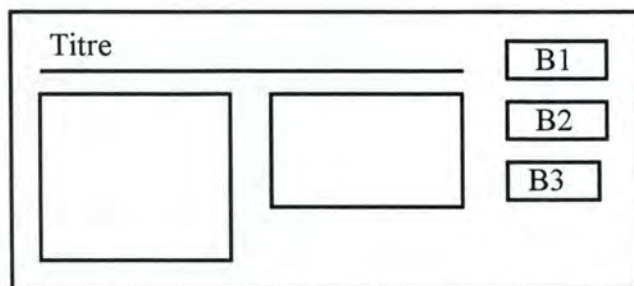
Les différentes étapes de l'algorithme sont :

- 1- caractérisation des objets
- 2- fixation des paramètres
- 3- calcul des mesures standards de la boîte
- 4- calcul des dimensions de colonnes
- 5- calcul de la proportion intérieure
- 6- ajout des boutons standards
- 7- disposition des boutons
- 8- choix des marges

Idée de base : construire une boîte suivant la grille ci-après.



Exemple :



IV.2.2 CARACTERISATION DES OBJETS

Les objets interactifs sont caractérisés par deux ensembles : \mathcal{H} et \mathcal{A} .

1°) \mathcal{H} = hiérarchie ordonnée des OIA

$\mathcal{H} = \{ h_i \},$

avec $h_i = (l_i, inv_i, oia_i, ld_i),$

où l_i = libellé identificatif,

inv_i = invitation à saisir,

oia_i = type d'objet interactif abstrait (champ d'édition unilinéaire, Boîte à ocher, ...)

et ld_i = libellé descriptif éventuel (Ex : m²).

Exemple :

Surface : (m²)

- - - ► $h = ("Surface", ":", EBX, "(m^2)")$

☒ Recto-verso

- - - ► $h = ("Recto-verso", "", CHX, "")$

Commande →

- - - ► $h = ("Commande", "→", EBX, "")$

où EBX = Edit BoX (abréviation pour champ d'édition unilinéaire),

CHX = CHeck boX (abréviation pour boîte à cocher).

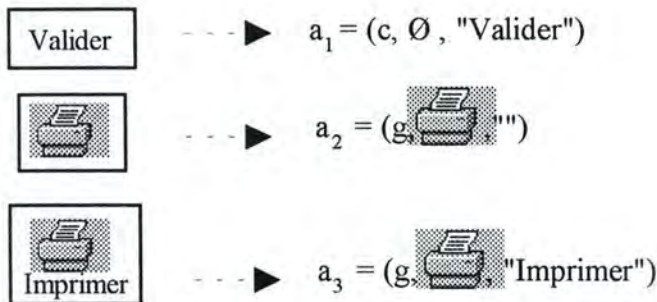
2°) \mathcal{A} = liste ordonnée des actions possibles pour un message externe.

Cette liste contient les fonctions sémantiques de l'application (telles que Valider, ...) et les fonctions de service locales (telles que Lister, Imprimer, ...)

$\mathcal{A} = \{a_i\}$, avec $a_i = (t_i, \text{ico}_i, \text{li}_i)$,

où t_i = type d'action (de commande (c) ou graphique (g)),
 ico_i = icône éventuelle du bouton graphique
 et li_i = libellé éventuel du bouton de commande.

Exemple :



IV.2.3 FIXATION DES PARAMETRES

Choix de U

Afin de déterminer les positions lors du placement, un système de référence à 2 dimensions basé sur des unités de mesure est nécessaire.

Soit U = unité de mesure.

Exemples d'unités : pixel, pica, caractère. Nous privilégions le pixel.

Choix de I

I = interligne standard

= nombre d'unités entre deux lignes.

I/2 = demi-interligne

Exemple de choix de I : 10 pixels.

Choix de H

H = hauteur standard

= nombre maximum d'unités pour une lettre majuscule.

Exemple de choix de H : 12 pixels.

Choix de L

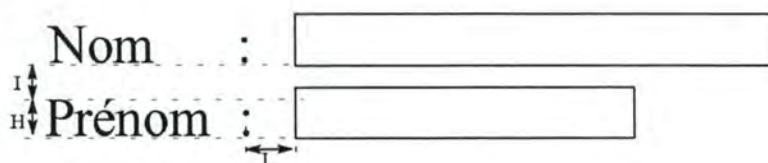
L = intercolonne standard

= nombre d'unités entre deux colonnes.

Exemple de choix de L : 4 pixels.

Les choix I = 10 U, H = 12 U et L = 40 U seront ceux retenus dans la suite du document.

Exemple :



IV.2.4 CALCUL DES MESURES STANDARDS

Calcul du nombre d'OIA

$$n = \#\mathcal{H}$$

Calcul du nombre d'actions

$$a = \#\mathcal{A}$$

Calcul des dimensions des OIC

Pour chaque h_i ,

avec $h_i = (l_i, inv_i, oia_i, ldi_i)$,

déterminer

$p_i = (l_{li}, pos_{li}, l_{invi}, l_{oiai}, h_{oiai}, l_{idi}, pos_{idi})$

et $q_i = (l_{itot}, h_{itot})$,

avec l_{li} = longueur du libellé identificatif,

pos_{li} = position du libellé identificatif = (gauche, au-dessus, droite),

l_{invi} = longueur de l'invitation à saisir éventuelle,

l_{oiai} = longueur de l'OIC proprement dit,

h_{oiai} = hauteur de l'OIC proprement dit,

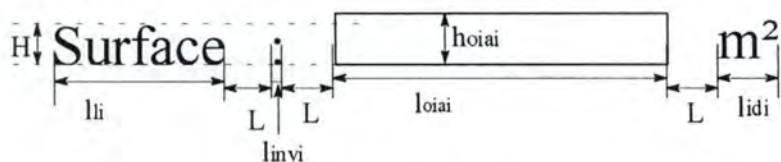
l_{idi} = longueur du libellé descriptif éventuel,

pos_{idi} = position du libellé descriptif éventuel = (gauche, droite, aucune),

l_{itot} = longueur totale de l'objet,

h_{itot} = hauteur totale de l'objet.

Exemples :



$pos_{li} = \text{gauche},$

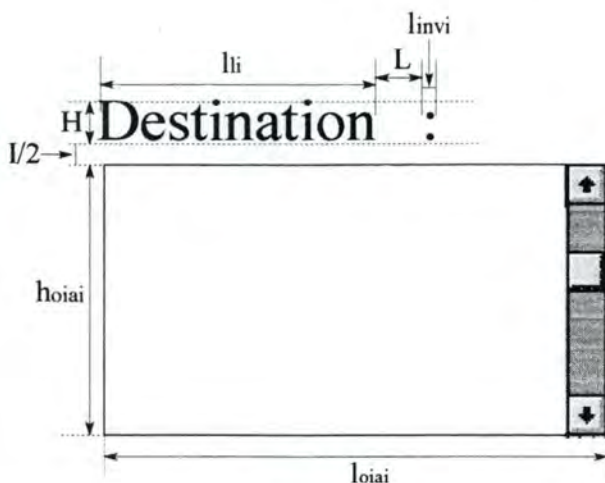
$pos_{idi} = \text{droite},$

$l_{itot} = l_{li} + L + l_{invi} + L + l_{oiai} + L + l_{idi},$

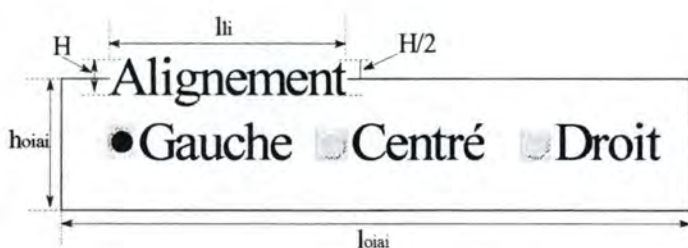
$h_{itot} = \max(h_{oiai}, H).$



$pos_{li} = \text{droite},$
 $pos_{idi} = \text{aucune},$
 $l_{invi} = 0,$
 $l_{idi} = 0,$
 $l_{itot} = l_{oiai} + L + l_{li},$
 $h_{itot} = H.$

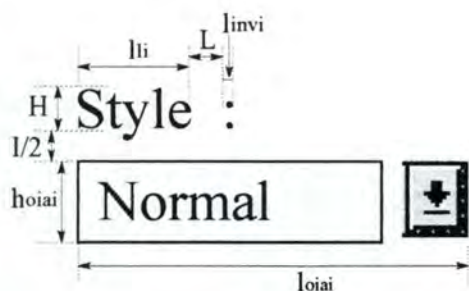


$pos_{li} = \text{au-dessus},$
 $pos_{idi} = \text{aucune},$
 $l_{idi} = 0,$
 $l_{itot} = \max(l_{oiai}, l_{li} + L + l_{invi}),$
 $h_{itot} = H + I/2 + h_{oiai}.$



$pos_{li} = \text{au-dessus},$
 $pos_{idi} = \text{aucune},$
 $l_{idi} = 0,$
 $l_{itot} = l_{oiai},$
 $h_{itot} = h_{oiai} + H/2.$

Remarque : h_{oiai} et l_{oiai} sont calculés à partir des hauteurs et longueurs des objets contenus dans la boîte de regroupement (ici, des items de bouton-radio).



$pos_{li} = \text{au-dessus},$
 $pos_{idi} = \text{aucune},$
 $l_{idi} = 0,$
 $l_{itot} = \max(l_{oiai}, l_{li} + L + l_{invi}),$
 $h_{itot} = H + I/2 + h_{oiai}.$

Bipartition des OIC

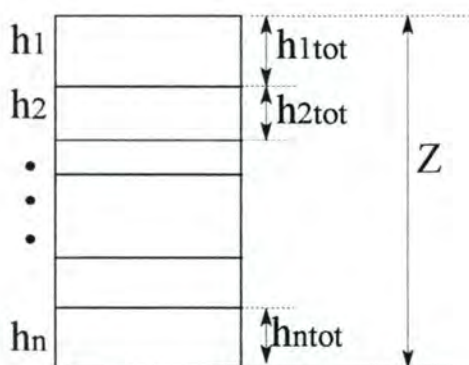
But

Répartir les OIC en deux colonnes équilibrées, c'est-à-dire de telle sorte que l'écart de hauteur entre les deux colonnes soit minimal.

Calculer Z

$$Z = \sum_{i=1}^n h_{itot}$$

= hauteur totale de tous les OIC empilés les uns au-dessus des autres.



Afin de répartir les OIA en deux colonnes équilibrées, on calcule le milieu $Z/2$ pour déterminer s'il tombe ou non dans un OIA. Deux cas se présentent, par conséquent, dans lesquels on détermine n_1 et n_2 qui sont respectivement le nombre d'OIA dans la première colonne et le nombre d'OIA dans la seconde colonne.

Si $Z/2$ tombe sur une frontière entre deux OIC

Il existe dans \mathcal{H} un objet de rang k tel que $\sum_{i=1}^k h_{itot} = Z/2.$

Posons $n_1 := k$ et $n_2 := n - n_1.$

Exemple :

h1	2	\Rightarrow	<table><tr><td>2</td></tr><tr><td>1</td></tr></table>	2	1	<table><tr><td>1</td></tr><tr><td>1</td></tr></table>	1	1
2								
1								
1								
1								
h2	1							
h3	1							
h4	1							
h5	1							

$$Z = 6 \Rightarrow Z/2 = 3$$

$$\text{et } k = 2, n_1 = 2 \text{ et } n_2 = 3.$$

Si $Z/2$ ne tombe pas sur une frontière entre deux OIC

$Z/2$ tombe dans le $k^{\text{ième}}$ OIC. Deux possibilités peuvent se présenter : bipartitionner au $(k-1)^{\text{ième}}$ objet (i.e. à l'objet précédent) ou bipartitionner au $k^{\text{ième}}$ objet (i.e. à l'objet courant). Pour décider cette bipartition, on calcule les écarts de longueurs des colonnes dans les deux sous-cas pour ne prendre que le meilleur agencement.

$$\square \text{ Calculer } \epsilon_1 = \left| Z/2 - \sum_{i=1}^{k-1} h_{i\text{tot}} \right|$$

$$\epsilon_2 = \left| \sum_{i=1}^k h_{i\text{tot}} - Z/2 \right|$$

$$\epsilon_{\min} = \min(\epsilon_1, \epsilon_2).$$

$$\square \text{ Si } \epsilon_1 \leq \epsilon_2, \text{ alors } n_1 = k - 1 \text{ (voir exemple 1)} \\ \text{sinon } n_1 = k \text{ (voir exemple 2).}$$

$$\square n_2 := n - n_1.$$

Exemple 1 :

h1	6
h2	2
h3	4
h4	1

$$Z = 13 \Rightarrow Z/2 = 6,5 \text{ et } k = 2.$$

$$\epsilon_1 = |6,5 - 6| = 0,5$$

$$\epsilon_2 = |8 - 6,5| = 1,5$$

$$\epsilon_1 \leq \epsilon_2 \Rightarrow n_1 = 1.$$

L'écart de distance est minimal pour l'OIA précédent (i.e. le $k^{\text{ième}}$ OIA). La bipartition obtenue sera :

6	2
	4
	1

Exemple 2 :

h1	2
h2	5
h3	4

$$Z = 11 \Rightarrow Z/2 = 5,5 \text{ et } k = 2.$$

$$\epsilon_1 = |5,5 - 2| = 3,5$$

$$\epsilon_2 = |7 - 5,5| = 1,5$$

$$\epsilon_1 > \epsilon_2 \Rightarrow n_1 = 2.$$

L'écart de distance est minimal pour l'OIA courant compris. La bipartition obtenue sera :

2	4
5	

Transfert éventuel du dernier OIC de droite à gauche

$$\text{Calculer } \epsilon_3 = \left| \sum_{i=1}^{n_1} h_{i\text{tot}} + h_{\text{mtot}} - \sum_{i=n_1+1}^{n-1} h_{i\text{tot}} \right|.$$

Si $\epsilon_3 < \epsilon_{\min}$, on peut faire passer le dernier OIC de droite vers la gauche. Il faudra alors réorganiser \mathcal{H} .

$$\mathcal{H} = \{h_1, h_2, \dots, h_{n_1}, h_n, h_{n_1+1}, \dots, h_{n-1}\}.$$

$$n_1 := n_1 + 1,$$

$$n_2 := n_2 - 1.$$

Exemple :

h1	2		
h2	4		
h3	1	2	4

$$Z = 7 \Rightarrow Z/2 = 3,5 \text{ et } k = 2.$$

$$\epsilon_1 = 1,5, \epsilon_2 = 2,5 \text{ et } \epsilon_3 = 1 < \epsilon_{\min}.$$

2	4
1	

Optimisation du placement des OIC au sein des deux colonnes

Heuristique de justification

S'il y a une sous-suite d'OIC identiques, leurs libellés et/ou leurs OI et/ou leurs invitations à saisir peuvent être justifier à gauche ou à droite.

Exemple 1 :

Nom : Nom :
Prénom : Prénom :

→

Exemple 2 :

Nom du programme : Nom du programme :
Etat : Etat :

→

si $\exists j, k$ tq $1 \leq j < k \leq n$ et $oia_j = oia_{j+1} = \dots = oia_k$

alors calculer $l_{\min} = \min_{i=j, \dots, k}(l_i)$

$$l_{\max} = \max_{i=j, \dots, k}(l_i)$$

si $l_{\min} \in [l_{\max} - 5, l_{\max}]$

alors $\forall i \in [j, k] : l_{li} = l_{\max}$

(* si l'écart des longueurs n'excède pas 5,
alors justification à gauche *)

sinon $\forall i \in [j, k] : l_{li} = (l_{\max} - l_{li})\text{blanc} + l_{li}$ (* sinon justification à droite *)

Heuristique d'alignement des libellés

Les libellés des boîtes de regroupement peuvent être justifiés à gauche avec le libellé le plus à gauche.

Exemple :

Prénom : Prénom :
Adresse Adresse

→

Heuristique de justification à droite

Tous les OIC d'une même colonne peuvent être justifiés à droite, si cela est permis.

Exemple :

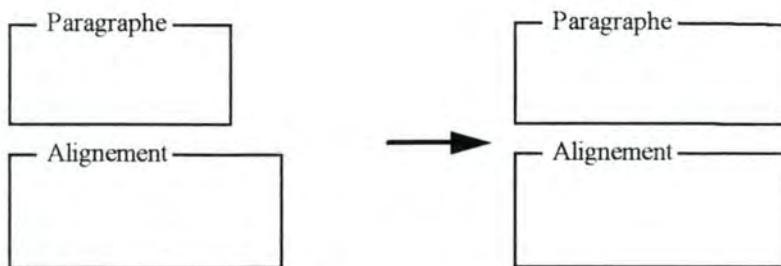
Nom : Nom :
Adresse Adresse

→

Heuristique d'uniformisation des boîtes de regroupement

Les boîtes de regroupement doivent être uniformisés au sein de chaque colonne.

Exemple :

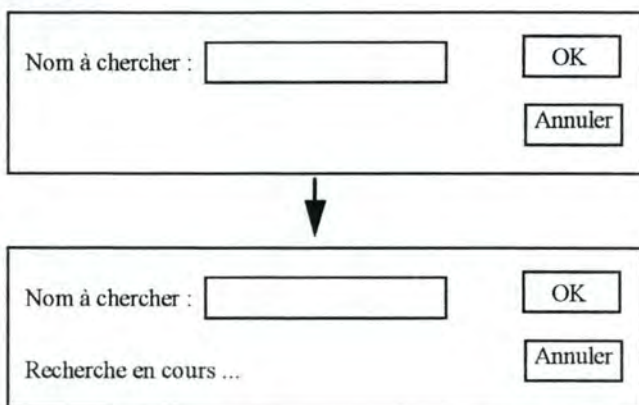


Ajout de séparateurs à la demande de l'utilisateur

Cette opération est permise dans l'éditeur graphique de l'interface en manipulation directe.

Ajout d'une zone de message à la demande de l'utilisateur

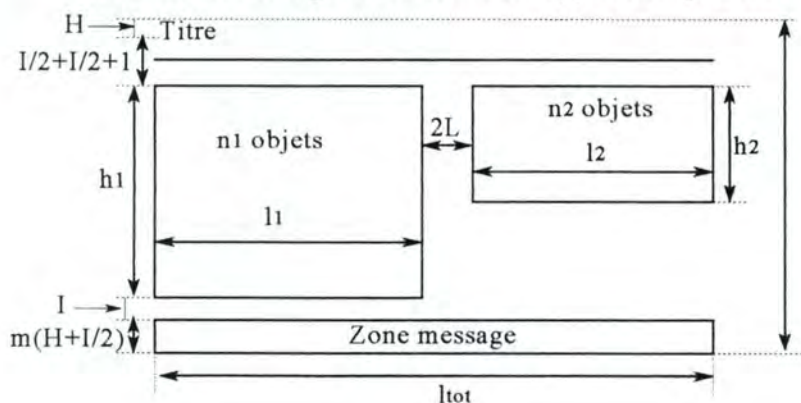
Exemple :



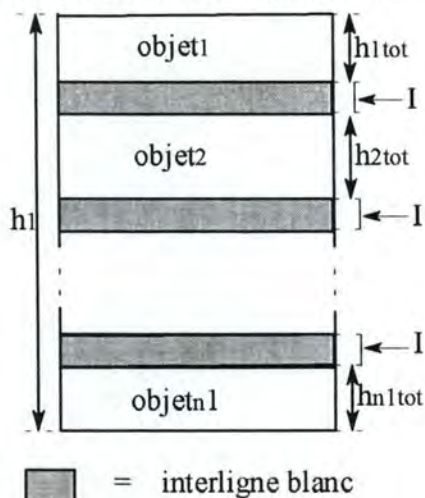
Ajout d'un titre souligné

IV.2.5 CALCUL DES DIMENSIONS DES COLONNES

La boîte de dialogue a maintenant l'apparence générale suivante :



La première colonne des OIC a la structure suivante :



Calcul des dimensions de la colonne 1

$$h_1 = \sum_{i=1}^{n_1} h_{itot} + (n_1 - 1)I,$$

$$l_1 = \max_{i=1, \dots, n_1}(l_{itot}).$$

Calcul des dimensions de la colonne 2

$$h_2 = \sum_{i=n_1+1}^n h_{itot} + (n_2 - 1)I,$$

$$l_2 = \max_{i=n_1+1, \dots, n}(l_{itot}).$$

Calcul des dimensions totales

$$h_{tot} = H + I/2 + 1 + I/2 + \max(h_1, h_2) + I + m(H + I/2),$$

$$l_{tot} = l_1 + 2L + l_1.$$

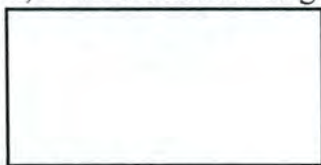
N.B. Le terme $I + m(H + I/2)$ de h_{tot} est à prendre en compte uniquement lorsqu'il y a m lignes de message ($m > 0$).

IV.2.6 CALCUL DE LA PROPORTION INTERIEURE

L'objectif est de calculer le rapport actuel entre la longueur totale et la hauteur totale afin de le faire se rapprocher autant que possible d'une proportion harmonieuse communément utilisée.

$$p_{int} = l_{tot} / h_{tot}.$$

□ Si $p_{int} \geq 1$, alors la boîte est du genre :



On est dans de bonnes conditions.

- Si $p_{int} < 1$, alors la boîte est du genre :



et il se peut qu'un re-placement soit nécessaire.

IV.2.7 AJOUT DES BOUTONS STANDARDS

But : Ajouter à \mathcal{A} , en respectant l'ordre ergonomique,

- de manière automatique : (OK), (Annuler)
- de manière assistée : (Appliquer), (Réinitialiser), (Aide), (A propos de ...)

$\Rightarrow \mathcal{A}_{nouv} = \{(OK), (Annuler), (Réinitialiser), (a_i \text{ anciens}), (Aide), (A propos de ...)\}$.

Le cas échéant recalculer

$$a = \#\mathcal{A}_{nouv}$$

et $c =$ nombre de boutons de commande,

$g =$ nombre de boutons graphiques.

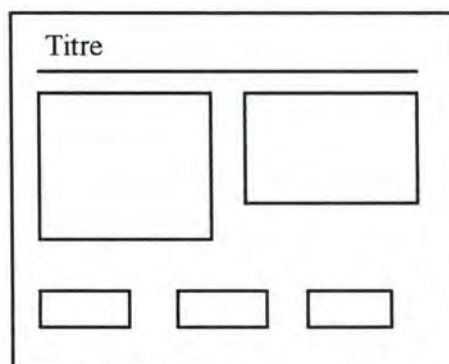
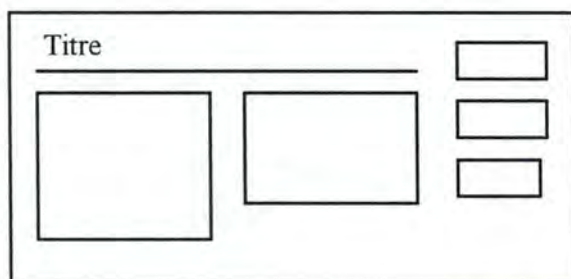
On doit avoir $a = c + g$.

IV.2.8 DISPOSITION DES BOUTONS

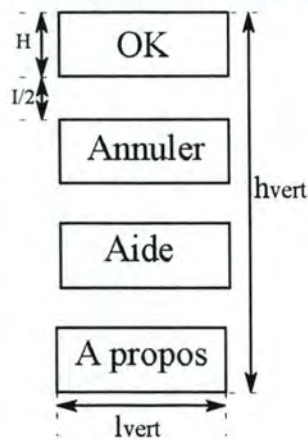
Soit \bar{H} la hauteur standard des boutons. On a $\bar{H} > H$.

L'objectif est d'adopter

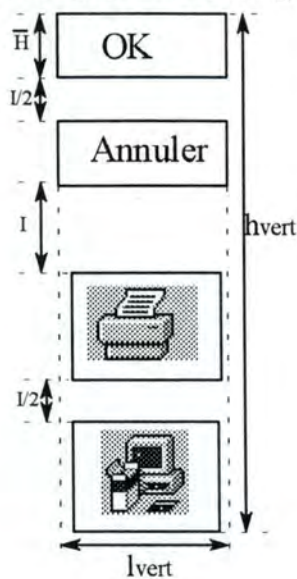
- soit une disposition verticale des boutons, si possible équilibrés en longueur ou proportionnellement,
- soit une disposition horizontale des boutons, si possible équilibrés en hauteur ou proportionnellement.



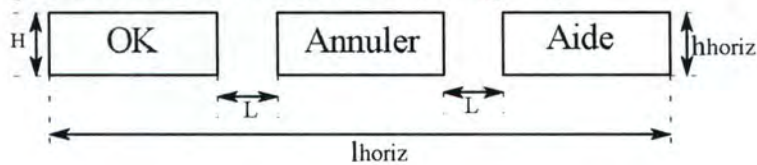
Situation générale de la disposition verticale
Avec uniquement des boutons de commande ($g = 0$)



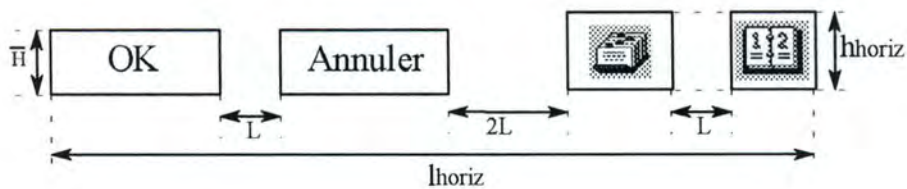
Avec boutons de commande et boutons graphiques (mixte, $g \neq 0$)



Situation générale de la disposition horizontale
Avec uniquement des boutons de commande ($g = 0$)



Avec boutons de commande et boutons graphiques (mixte, $g \neq 0$)



Calculer les hauteur et longueur dans le cas de la disposition verticale

$$h_{\text{vert}} = c * \bar{H} + (c - 1) * I/2 \quad \text{si } g = 0,$$

$$= c * \bar{H} + (c - 1) * I/2 + I + (g - 1) * I/2 + \sum_{i=1}^q h_{\text{itot}} \quad \text{si } g \neq 0,$$

$$l_{\text{vert}} = \max_{i=1, \dots, c} (l_{\text{itot}}) + 2L \quad \text{si } g = 0,$$

$$= \max [\max_{i=1, \dots, c} (l_{\text{itot}}) + 2L, \max_{i=1, \dots, g} (l_{\text{itot}})] \quad \text{si } g \neq 0,$$

où l_{itot} et h_{itot} sont la longueur totale et la hauteur totale du bouton i .

Calculer les hauteur et longueur dans le cas de la disposition horizontale

$$h_{\text{horiz}} = \bar{H} \quad \text{si } g = 0,$$

$$= \max [\bar{H}, \max_{i=1, \dots, g} (h_{\text{itot}})] \quad \text{si } g \neq 0,$$

$$l_{\text{horiz}} = c * \max_{i=1, \dots, c} (l_{\text{itot}}) + (c - 1) * L \quad \text{si } g = 0,$$

$$= c * \max_{i=1, \dots, c} (l_{\text{itot}}) + (c - 1) * L + 2L + \sum_{i=1}^q l_{\text{itot}} + (g - 1) * L \quad \text{si } g \neq 0,$$

où l_{itot} et h_{itot} sont la longueur totale et la hauteur totale du bouton i .

Heuristique de placement

si $p_{\text{int}} \geq 1$

alors si $h_{\text{vert}} \leq h_{\text{tot}}$

alors disposer le groupe de boutons verticalement à droite (cas 1)

sinon si $l_{\text{horiz}} \leq l_{\text{tot}}$

alors disposer les boutons horizontalement en bas (cas 2)

sinon si $g \neq 0$

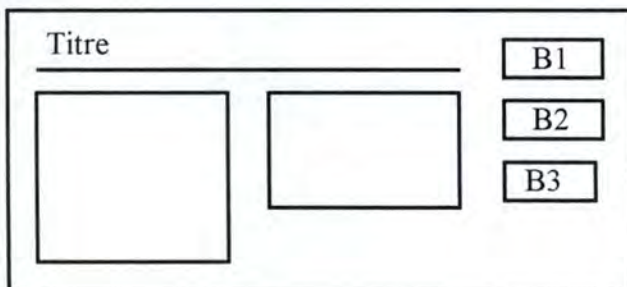
alors disposer les boutons en deux groupes verticaux à droite (cas 3)

sinon disposer les boutons verticalement à droite (cas 4)

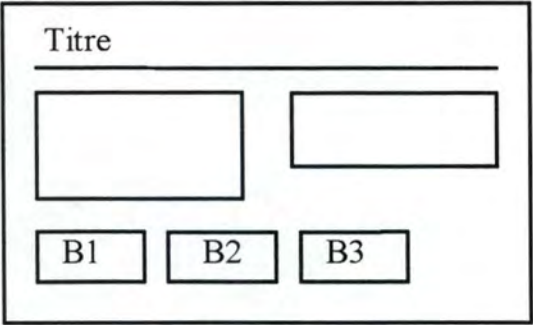
sinon disposer les boutons verticalement à droite (cas 5)

Illustration des 5 cas :

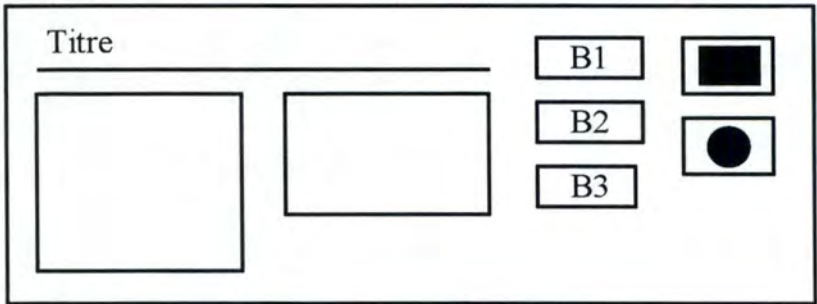
cas 1



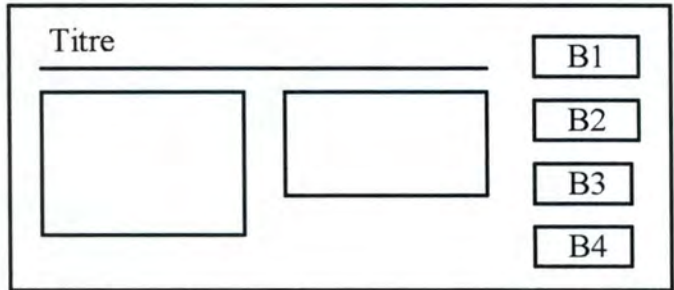
cas 2



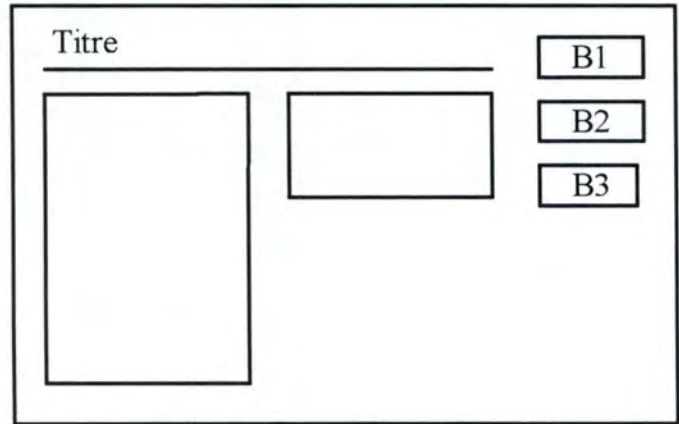
cas 3



cas 4



cas 5



Recalcul des dimensions totales

Placement (cas 1) : h_{tot} inchangé,
 $l_{tot} := l_{tot} + 2L + l_{vert}$.

Placement (cas 2) : $h_{tot} := h_{tot} + I + h_{horiz}$,
 l_{tot} inchangé.

Placement (cas 3) : $h_{tot} := \max [h_{tot}, h_{tot}(g), h_{tot}(c)]$,
 $l_{tot} := l_{tot} + 2L + l_{vert}(c) + 2L + l_{vert}(g)$.

Placement (cas 4) : $h_{tot} := h_{vert}$,
 $l_{tot} := l_{tot} + 2L + l_{vert}$.

Placement (cas 5) : h_{tot} inchangé,
 $l_{tot} := l_{tot} + 2L + l_{vert}$.

IV.2.9 CHOIX DES MARGES

Soit $M = \{ 1:1, 1:1,29, 1:1,33, 1:1,4142, 1:1,5, 1:1,6, 1:1,618, 1:1,7321, 1:2 \}$.

M représente l'ensemble des proportions considérées comme ergonomiquement valables.

Calculer la nouvelle proportion

$$p_{int} = l_{tot} / h_{tot}.$$

Si $p_{int} < 1$,

alors réorganiser la boîte,

sinon $\exists m \in M$ tq $p_{int} \leq m = \min(\{x \in M : x \geq p_{int}\})$,

soit $h_{final} := h_{tot} + 2I$,

calculer $l_{final} := h_{final} * m$,

(l_{final}, h_{final}) donne les dimensions extérieures de la boîte de dialogue.

IV.3 APPLICATION DE L'ALGORITHME À UN EXEMPLE

IV.3.1 CARACTÉRISATION DES OBJETS

Nous considérons comme exemple, la réalisation d'une interface sur l'admission d'un patient dans un hôpital. Cet exemple est issu des travaux pratiques du cours intitulé : méthodologie de développement des logiciels.

Caractérisation des objets

$$\mathcal{H}_{admission} = \{ h_{dj}, h_{pat}, h_{co}, h_{numtit}, h_{typeaf}, h_{méd}, h_{serv}, h_{cat}, h_{rég} \},$$

avec $h_i = (l_i, inv_i, oia_i, ld_i)$,

où l_i = libellé identificatif,

inv_i = invitation à saisir,

oia_i = type d'objet interactif abstrait (champ d'édition unilinéaire, Boîte à cocher, .)

et ld_i = libellé descriptif éventuel (Ex : m²).

$h_{dj} = (\text{"Date du jour"}, ":", \text{EBX}, \text{""})$,

$h_{pat} = (\text{"Patient"}, "", \text{GBX}, \text{""})$,

$h_{co} = (\text{"Code de l'organisme", ":", EBX, ""}),$
 $h_{numtit} = (\text{"Numéro de titulaire", ":", EBX, ""}),$
 $h_{typeaf} = (\text{"Type d'affiliation", ":", EBX, ""}),$
 $h_{méd} = (\text{"Médecin", ":", EBX, ""}),$
 $h_{serv} = (\text{"Service", ":", SLB, ""}),$
 $h_{cat} = (\text{"Catégorie de chambre", ":", GBX, ""}),$
 $h_{rég} = (\text{"Régime", ":", SLB, ""}).$

$\mathcal{H}_{pat} = \{h_{p1}, h_{p2}, h_{p3}, h_{p4}, h_{p5}, h_{p6}, h_{p7}\},$
 $h_{p1} = (\text{"Nom", ":", EBX, ""}),$
 $h_{p2} = (\text{"Prénom", ":", EBX, ""}),$
 $h_{p3} = (\text{"Date de naissance", ":", EBX, ""}),$
 $h_{p4} = (\text{"Adresse", ":", EBX, ""}),$
 $h_{p5} = (\text{"Téléphone", ":", EBX, ""}),$
 $h_{p6} = (\text{"Sexe", "", GBX, ""}),$
 $h_{p7} = (\text{"Etat civil", "", GBX, ""}).$

$\mathcal{H}_{cat} = \{hc1, hc2, hc3\},$
 $hc1 = (\text{"Chambre particulière", "", RBX, ""}),$
 $hc2 = (\text{"Chambre à 2 lits", "", RBX, ""}),$
 $hc3 = (\text{"Chambre à 4 lits", "", RBX, ""}).$

$\mathcal{H}_{sexe} = \{h_{s1}, h_{s2}\},$
 $h_{s1} = (\text{"Masculin", "", RBX, ""}),$
 $h_{s2} = (\text{"Féminin", "", RBX, ""}).$

$\mathcal{H}_{ec} = \{h_{ec1}, h_{ec2}, h_{ec3}, h_{ec4}\},$
 $h_{ec1} = (\text{"Célibataire", "", RBX, ""}),$
 $h_{ec2} = (\text{"Marié", "", RBX, ""}),$
 $h_{ec3} = (\text{"Veuf", "", RBX, ""}),$
 $h_{ec4} = (\text{"Divorcé", "", RBX, ""}).$

Remarque :

EBX = Champ d'édition unilinéaire,
 SLB = Liste de sélection déroulante,
 GBX = Boîte de regroupement,
 RBX = Bouton-radio.

IV.3.2 FIXATION DES PARAMÈTRES

Nous avons préféré prendre U comme unité de longueur pour les dessins au lieu du pixel car cela était plus pratique.

U = 1 unité.

Interligne standard : I = U.

Hauteur standard : H = 1,5U.

Intercolonne standard : L = 0,5U.

Diamètre d'un OI d'item de bouton-radio = U.
 $L < I \leq H$.

IV.3.3 CALCUL DES MESURES STANDARDS

$n = \# \mathcal{H}_{\text{admission}}$ = nombre des OIA dans la boîte de dialogue Admission

$a = \# \mathcal{A}$ = nombre d'actions pour un message externe

$a = 0$ dans notre cas.

Calcul des dimensions des OIC

Soient les vecteurs p_i et q_i tels que

$p_i = (l_{li}, pos_{li}, l_{invi}, l_{oiai}, h_{oiai}, l_{idi}, pos_{idi})$

et $q_i = (l_{itot}, h_{itot})$,

Calcul des coordonnées des objets composant Etat-civil

Célibataire :

$p_{ec1} = (l_{li} = 5,5, pos_{li} = \text{droite}, l_{invi} = 0, l_{oiai} = 1, h_{oiai} = 1,5, l_{idi} = 0, pos_{idi} = \text{aucune}),$
 $q_{ec1} = (l_{itot} = l_{oiai} + L + l_{li} = 7, h_{itot} = 1,5).$

Marié :

$p_{ec2} = (l_{li} = 2,5, pos_{li} = \text{droite}, l_{invi} = 0, l_{oiai} = 1, h_{oiai} = 1,5, l_{idi} = 0, pos_{idi} = \text{aucune}),$
 $q_{ec2} = (l_{itot} = 4, h_{itot} = 1,5).$

Veuf :

$p_{ec3} = (l_{li} = 2, pos_{li} = \text{droite}, l_{invi} = 0, l_{oiai} = 1, h_{oiai} = 1,5, l_{idi} = 0, pos_{idi} = \text{aucune}),$
 $q_{ec3} = (l_{itot} = 3,5, h_{itot} = 1,5).$

Divorcé :

$p_{ec4} = (l_{li} = 3,5, pos_{li} = \text{droite}, l_{invi} = 0, l_{oiai} = 1, h_{oiai} = 1,5, l_{idi} = 0, pos_{idi} = \text{aucune}),$
 $q_{ec4} = (l_{itot} = 5, h_{itot} = 1,5).$

D'où,

Etat-civil :

$q_{ec} = (l_{itot} = 7, h_{itot} = 1,5*4 + 3*I = 9).$

On en déduit pour \mathcal{H}_{ec}

$l_{itot} = (\text{distance avec le cadre}) + 7$
 $= 2L + 7$
 $= 8, (\text{car } L = 1/2)$

$h_{itot} = 9 + (\text{hauteur de Etat-civil}) + (\text{distance entre Etat-civil et Célibataire}) +$
 $(\text{distance entre Divorcé et le bord inférieur du cadre})$
 $= 9 + H + I + I/2$
 $= 12,$

soit $q_{p7} = (l_{itot} = 8, h_{itot} = 12).$

Calcul des dimensions des objets composant Sexe

Masculin :

$$p_{s1} = (l_{li} = 4, \text{pos}_{li} = \text{droite}, l_{invi} = 0, l_{oiai} = 1, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{s1} = (l_{itot} = l_{oiai} + L + l_{li} = 5,5, h_{itot} = 1,5)$$

Féminin :

$$p_{s2} = (l_{li} = 3,5, \text{pos}_{li} = \text{droite}, l_{invi} = 0, l_{oiai} = 1, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{s2} = (l_{itot} = 5, h_{itot} = 1,5).$$

D'où,

Sexe :

$$q_{sexe} = (l_{itot} = 5,5, h_{itot} = 1,5 * 2 + I = 4).$$

On en déduit pour \mathcal{H}_{sexe} :

$$l_{itot} = \text{distance avec le cadre} + 5,5$$

$$= 2L + 5,5$$

$$= 6,5,$$

$$h_{itot} = 4 + (\text{hauteur de Etat-civil}) + (\text{distance entre Etat-civil et Célibataire}) +$$

$$(\text{distance entre Divorcé et le bord inférieur du cadre})$$

$$= 4 + H + I + I/2$$

$$= 7,$$

$$\text{soit } q_{p6} = (l_{itot} = 6,5, h_{itot} = 7).$$

Calcul des dimensions des autres objets de la boîte de regroupement

Nom :

$$p_{p1} = (l_{li} = 1,5, \text{pos}_{li} = \text{gauche}, l_{invi} = 0,5, l_{oiai} = 10, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{p1} = (l_{itot} = l_{oiai} + L + l_{li} + L = 13, h_{itot} = 1,5).$$

Prénom :

$$p_{p2} = (l_{li} = 3, \text{pos}_{li} = \text{gauche}, l_{invi} = 0,5, l_{oiai} = 10, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{p2} = (l_{itot} = l_{oiai} + L + l_{li} + L = 14,5, h_{itot} = 1,5).$$

Date de naissance :

$$p_{p3} = (l_{li} = 8,5, \text{pos}_{li} = \text{gauche}, l_{invi} = 0,5, l_{oiai} = 4, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{p3} = (l_{itot} = 14, h_{itot} = 1,5).$$

(On suppose que la date se met sous la forme jj:mm:aa. D'où $l_{oiai} = 4$).

Adresse :

$$p_{p4} = (l_{li} = 3,5, \text{pos}_{li} = \text{gauche}, l_{invi} = 0,5, l_{oiai} = 15, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{p4} = (l_{itot} = 20, h_{itot} = 1,5).$$

Téléphone :

$$p_{p5} = (l_{li} = 4,5, \text{pos}_{li} = \text{gauche}, l_{invi} = 0,5, l_{oiai} = 6, h_{oiai} = 1,5, l_{idi} = 0, \text{pos}_{idi} = \text{aucune}),$$

$$q_{p5} = (l_{itot} = 12, h_{itot} = 1,5).$$

Boîte de regroupement Patient

Patient :

$$l_{itot} = (\text{distance avec le cadre}) + \max(l_{li}) + L + l_{invi} + L + \max(l_{oiai}) + L$$

$$= 2L + 8,5 + 3L + 0,5 + 15$$

$$= 26,5,$$

$$\begin{aligned}
h_{\text{itot}} &= 1,5*5 + (\text{hauteur de Patient}) + (\text{distance entre Patient et Nom}) + I*6 + \\
&\quad (\text{distance entre les bords inférieurs des cadres de Patient et de Etat-civil}) + \\
&\quad h_{\text{sexe}} + h_{\text{ec}} \\
&= 1,5*5 + H + I + I*6 + I/2 + h_{\text{sexe}} + h_{\text{ec}} \\
&= 35,5,
\end{aligned}$$

$$\text{soit } q_{\text{pat}} = (l_{\text{itot}} = 26,5, h_{\text{itot}} = 35,5).$$

Calcul des dimensions des objets restants

Date du jour :

$$\begin{aligned}
p_{\text{dj}} &= (l_{\text{li}} = 6, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 4, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{dj}} &= (l_{\text{itot}} = 11,5, h_{\text{itot}} = 1,5).
\end{aligned}$$

Code de l'organisme :

$$\begin{aligned}
p_{\text{co}} &= (l_{\text{li}} = 9,5, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 3, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{co}} &= (l_{\text{itot}} = 14, h_{\text{itot}} = 1,5).
\end{aligned}$$

Numéro de titulaire :

$$\begin{aligned}
p_{\text{numtit}} &= (l_{\text{li}} = 9,5, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 3, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{numtit}} &= (l_{\text{itot}} = 14, h_{\text{itot}} = 1,5).
\end{aligned}$$

Type d'affiliation :

$$\begin{aligned}
p_{\text{typeaf}} &= (l_{\text{li}} = 9, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 2, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{typeaf}} &= (l_{\text{itot}} = 12,5, h_{\text{itot}} = 1,5).
\end{aligned}$$

Médecin :

$$\begin{aligned}
p_{\text{méd}} &= (l_{\text{li}} = 3,5, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 3, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{méd}} &= (l_{\text{itot}} = 8, h_{\text{itot}} = 1,5).
\end{aligned}$$

Service :

$$\begin{aligned}
p_{\text{serv}} &= (l_{\text{li}} = 3,5, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 10, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{serv}} &= (l_{\text{itot}} = l_{\text{li}} + L + l_{\text{invi}} + L + l_{\text{oi}} = 15, h_{\text{itot}} = 1,5).
\end{aligned}$$

Régime :

$$\begin{aligned}
p_{\text{rég}} &= (l_{\text{li}} = 3, \text{pos}_{\text{li}} = \text{gauche}, l_{\text{invi}} = 0,5, l_{\text{oi}} = 10, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{rég}} &= (l_{\text{itot}} = 14,5, h_{\text{itot}} = 1,5).
\end{aligned}$$

Boîte de regroupement Catégorie de chambre

Chambre particulière :

$$\begin{aligned}
p_{\text{c1}} &= (l_{\text{li}} = 10, \text{pos}_{\text{li}} = \text{droite}, l_{\text{invi}} = 0, l_{\text{oi}} = 1, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{c1}} &= (l_{\text{itot}} = 11,5, h_{\text{itot}} = 1,5).
\end{aligned}$$

Chambre à 2 lits :

$$\begin{aligned}
p_{\text{c2}} &= (l_{\text{li}} = 8, \text{pos}_{\text{li}} = \text{droite}, l_{\text{invi}} = 0, l_{\text{oi}} = 1, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{c2}} &= (l_{\text{itot}} = 9,5, h_{\text{itot}} = 1,5).
\end{aligned}$$

Chambre à 4 lits :

$$\begin{aligned}
p_{\text{c3}} &= (l_{\text{li}} = 8, \text{pos}_{\text{li}} = \text{droite}, l_{\text{invi}} = 0, l_{\text{oi}} = 1, h_{\text{oi}} = 1,5, l_{\text{idi}} = 0, \text{pos}_{\text{idi}} = \text{aucune}), \\
q_{\text{c3}} &= (l_{\text{itot}} = 9,5, h_{\text{itot}} = 1,5).
\end{aligned}$$

D'où,

Catégorie de chambre :

$$q_{\text{cc}} = (l_{\text{itot}} = 11,5, h_{\text{itot}} = 1,5*3 + 2*I = 6,5).$$

On en déduit pour \mathcal{H}_{cat} :

$$\begin{aligned} l_{itot} &= (\text{distance avec le cadre}) + 11,5 \\ &= 2L + 11,5 \\ &= 12,5, \end{aligned}$$

$$\begin{aligned} h_{itot} &= 1,5*3 + (\text{hauteur de Catégorie de Chambre}) + (\text{distance entre Catégorie de chambre et Chambre particulière}) + I*2 + (\text{distance entre le bord inférieur du cadre et chambre à 4 lits}) \\ &= 1,5*3 + H + I + I*2 + I/2 \\ &= 9,5, \end{aligned}$$

$$\text{soit } q_{cat} = (l_{itot} = 12,5, h_{itot} = 9,5).$$

IV.3.4 CALCUL DE Z

$$Z = \sum_{i=1}^n h_{itot} = 1,5 + 35,5 + 6*1,5 + 9,5 = 55,5$$

(Ici on ne tient pas compte des espaces I entre les OIC)

$Z/2 = 27,75$: on se situe dans la boîte de regroupement Patient.

$$\varepsilon_1 = 27,75 - 1,5 = 26,25$$

$$\varepsilon_2 = 1,5 + 35,5 - 27,75 = 9,25$$

$$\varepsilon_1 > \varepsilon_2 \Rightarrow n = k = 2.$$

La première colonne va donc jusqu'à la fin de Patient.

IV.3.5 CALCUL DES DIMENSIONS DES COLONNES

Hauteur de la première colonne :

$$\begin{aligned} h_1 &= 1,5 + 35,5 + I \\ &= 1,5 + 35,5 + 1 \\ &= 38. \end{aligned}$$

Hauteur de la deuxième colonne :

$$\begin{aligned} h_2 &= 1,5*6 + 9,5 + 6*I \\ &= 1,5*6 + 9,5 + 6*1 \\ &= 24,5. \end{aligned}$$

Hauteur totale de la boîte de dialogue :

$$\begin{aligned} h_{tot} &= (\text{hauteur de Admission}) + (\text{distance entre le trait et Admission}) + (\text{distance entre Date du jour et le trait}) + 38 + (\text{distance entre la Boîte Patient et le message}) + (\text{hauteur de Message}) \\ &= H + I/2 + I/2 + 38 + I + H \\ &= 43 \end{aligned}$$

Longueur totale de la boîte de dialogue :

$$\begin{aligned} l_{tot} &= (\text{longueur de la première colonne} + (\text{longueur de la deuxième colonne} + (\text{distance entre les deux colonne})) \\ &= l_1 + l_2 + 2L, \end{aligned}$$

$$\begin{aligned} \text{avec } l_1 &= \text{longueur de la boîte de regroupement Patient} \\ &= 26,5 \end{aligned}$$

$$\begin{aligned}
 l_2 &= (\text{longueur de l'élément le plus long}) + L \\
 &= l_{\text{serv}} + L \\
 &= 15,5.
 \end{aligned}$$

Ainsi $l_{\text{tot}} = l_1 + l_2 + 2L$

$$\begin{aligned}
 &= 26,5 + 15,5 + 1 + 0,5 \\
 &= 43,5.
 \end{aligned}$$

Calcul de la proportion interne

$$p_{\text{int}} = l_{\text{tot}} / h_{\text{tot}} = 43,5/43 = 1,012.$$

IV.3.6 AJOUT DES BOUTONS STANDARDS

$$A_{\text{nouv}} = \{ (\text{OK}), (\text{Annuler}) \}.$$

Disposition des boutons2

Il n'existe pas de boutons graphiques. Donc $g = 0$.

Nous fixons à 2 unités la hauteur de chaque bouton.

La longueur de chaque bouton est :
longueur de Annuler + $2L = 4,5$.

- a) Lorsque l'on choisit de disposer verticalement les boutons :

$$\begin{aligned}
 h_{\text{vert}} &= \text{Hauteur des boutons} + \text{hauteur des interlignes} \\
 &= 2*2 + I/2 \\
 &= 4,5,
 \end{aligned}$$

$$\begin{aligned}
 l_{\text{vert}} &= \text{Longueur du plus long libellé} + 2L \\
 &= 3,5 + 1 \\
 &= 4,5.
 \end{aligned}$$

- b) Lorsque l'on choisit de disposer horizontalement les boutons :

$$\begin{aligned}
 h_{\text{horiz}} &= \text{hauteur d'un bouton} \\
 &= 2,
 \end{aligned}$$

$$\begin{aligned}
 l_{\text{horiz}} &= \text{longueur de tous les boutons} + \text{longueur des intercolonnes} \\
 &= 2(3,5 + 1) + L \\
 &= 9,5.
 \end{aligned}$$

Heuristique de placement des boutons :

$p_{\text{int}} = 1,012 > 1$. Nous nous trouvons donc dans la situation (cas 1) de l'algorithme (voir algorithme).

Recalcul des dimensions totales :

Placement (cas 1)

$$\begin{aligned}
 h_{\text{tot}} &= 43 \quad (h_{\text{tot}} \text{ est inchangé}), \\
 l_{\text{tot}} &= l_{\text{tot}} + 2L + l_{\text{vert}} = 43,5 + 1 + 4,5 = 49, \\
 p_{\text{int}} &= 49/43 = 1,139.
 \end{aligned}$$

Choix des marges :

Nous ajoutons $2L$ pour la marge gauche, $2L$ pour la marge droite, I pour la marge du haut et I pour la marge du bas.

En tenant compte de certaines critiques indispensables formulées dans la partie IV.4.1 (justification des boîtes de regroupement) nous obtenons comme résultat l'interface suivante :

Admission

Date du jour :

Patient

Nom :

Prénom :

Date de Naissance :

Adresse :

Téléphone :

Sexe

☒ Masculin

☐ Féminin

Etat civil

☒ Célibataire

☐ Marié

☐ Veuf

☐ Divorcé

Code de l'organisme :

Numéro de Titulaire :

Type d'affiliation :

Médecin :

Service : ▼

Catégorie de chambre

☒ Chambre particulière

☐ Chambre à 2 lits

☐ Chambre à 4 lits

Régime : ▼

Ok

Annuler

Nous remarquons que beaucoup d'espaces libres restent non utilisés notamment dans la boîte de regroupement *Patient*. Un arrangement horizontal des items de bouton-radio dans la boîte de regroupement Sexe et un arrangement sur deux lignes de deux items chacune dans la boîte de regroupement Etat civil aurait permis de réduire la hauteur de la boîte de regroupement Patient. On aurait ainsi eu moins d'espaces libres non utilisés.

IV.4 COMPARAISON DES ALGORITHMES DON ET TRIDENT

IV.4.1 SOUPLESSE DES DEUX STRATÉGIES

Les deux algorithmes utilisent un arbre qui donne la hiérarchie des objets interactifs concrets qui apparaîtront dans la boîte de dialogue. Toutefois, TRIDENT tient compte d'un ordre (prédéfini à la phase de sélection) pour les noeuds fils d'un même noeud père. Ces objets sont alors placés les uns après les autres en respectant l'ordre des noeuds pères et des noeuds fils. Dans DON, le placement se fait en considérant la similarité des noeuds fils.

Avantage de la stratégie DON

Le système génère plusieurs versions d'une même interface (une gamme entière de placements différents), et il est possible d'en choisir une afin de la raffiner interactivement. Il est, par exemple, possible de remplacer un objet interactif par un autre ayant la même sémantique.

Inconvénient de la stratégie

Etant donné qu'aucun ordre des objets interactifs concrets d'un même nœud n'est établi, on peut avoir une incohérence dans la suite logique des objets. En effet, il se

pourrait par exemple, que le nom et le prénom d'une personne dans une boîte de dialogue soient séparés par sa date de naissance.

Avantage de la stratégie TRIDENT

L'ordre logique des éléments est respecté dans la boîte de dialogue.

Inconvénients de la stratégie

Le système est rigide. Il ne propose qu'une seule version de l'interface, et il n'est pas possible de raffiner celle-ci de façon interactive.

IV.4.2 ANALYSE DES FORMES ET DES DIMENSIONS

Cette analyse est absente dans la stratégie TRIDENT.

Avantages de la stratégie

Le placement conduit à une interface plus esthétique. En outre, il minimise l'espace non utilisé.

Inconvénient de la stratégie

Des objets logiquement liés peuvent être séparés tout simplement parce que leurs formes ne sont pas similaires. Ce qui peut quelquefois conduire à des incohérences dans la suite logique des objets de l'interface.

IV.4.3 SOLUTIONS D'OVERFLOW

Dans TRIDENT les problèmes d'overflow sont résolus au niveau de la phase de sélection. Ceci pour éviter d'avoir toute situation conflictuelle future.

Avantage de la stratégie DON

Les solutions d'overflow sont envisagées uniquement quand le problème se pose. Dans beaucoup de cas, il suffira de jouer sur l'espacement et la taille des objets interactifs et sur les marges pour résoudre le problème sans pour autant reconsidérer toute l'interface.

Inconvénient de la stratégie

Si le problème d'overflow ne peut être résolu en jouant sur l'espacement et la taille des objets interactifs et sur les marges, la stratégie est obligée de repartir à la phase de sélection des objets interactifs, ce qui entraîne la construction d'un nouvel arbre de structuration des noeuds. Il y aurait donc eu beaucoup de travail pour rien.

Avantage de la stratégie TRIDENT

La stratégie garantit que pendant la phase de placement des objets interactifs concrets, on ne pourra avoir des problèmes d'overflow qui obligeraient à repartir à la phase de sélection.

Inconvénient

Le fait de résoudre les problèmes d'overflow dès la phase de sélection amène à être prudent dès que le nombre d'objets d'une boîte de dialogue est grand. Il peut donc arriver

que l'algorithme choisisse d'utiliser deux boîtes de dialogue au lieu d'une seule même lorsque cela n'est pas nécessaire.

IV.4.4 PLACEMENT DES BOUTONS

Nous n'avons pas d'informations sur la façon dont les boutons autres que "OK" et "CANCEL" sont disposés dans les boîtes de dialogue générées par DON.

Avantage de la stratégie DON

L'utilisation du "remainder area" (espace non utilisé en bas et à droite de l'interface après le placement de tous les objets interactifs exceptés les boutons "OK" et "CANCEL"), pour le placement des boutons "OK" et "CANCEL" lorsque cela est possible, permet de minimiser les dimensions de l'interface. Il en résulte aussi que l'interface est plus équilibrée et plus esthétique.

Inconvénient

Le fait que l'on puisse avoir les boutons "OK" et "CANCEL" suivant diverses localisations et diverses orientations peut ne pas aider à un apprentissage aisé de l'application pour laquelle l'interface est générée. Il est cependant possible d'utiliser les profils de préférences pour obtenir une disposition unique des boutons "OK" et "CANCEL" à travers toute l'application.

Avantages de la stratégie TRIDENT

Les boutons sont toujours disposés horizontalement en bas ou verticalement à droite de l'interface ; ce qui permet un apprentissage plus facile. De plus, tous les boutons sont localisés au même endroit.

Inconvénients

Même si un grand espace reste inutilisé en dessous de la deuxième colonne de l'interface, il n'est pas considéré pour le placement des boutons. Il en résulte un accroissement des dimensions de l'interface et un déséquilibre au niveau du placement des objets.

IV.4.5 PLACEMENT EN DEUX COLONNES

L'algorithme TRIDENT répartit systématiquement les objets interactifs d'une boîte de dialogue (autres que les boutons) en deux colonnes.

Avantages de la stratégie

Quand le nombre d'objets interactifs dans la boîte de dialogue est élevé, son aspect visuel s'en trouve amélioré, vu les propriétés du bi-colonnage.

Inconvénient

Quand le nombre d'objets interactifs est peu élevé, on obtient une boîte de dialogue trop plate et peu esthétique.

IV.4.6 PROPORTION INTERNE

L'utilisation de la proportion intérieure est absente dans DON.

Avantage

L'utilisation de la proportion intérieure permet d'accroître la largeur de la boîte de dialogue lorsqu'elle est très haute (et peu large) en plaçant les boutons verticalement à droite.

Inconvénient

Dans beaucoup de cas la proportion intérieure est supérieure à 1 et la hauteur totale des boutons est inférieure à la hauteur de la plus grande colonne : l'algorithme placera les boutons verticalement à droite ; ce qui pourrait entraîner un accroissement de la largeur de la boîte. Cela peut conduire à une interface visuellement peu esthétique.

IV.5 CRITIQUES ET AMÉLIORATIONS DE L'ALGORITHME

Nous regroupons ces critiques selon un ordre de priorité.

IV.5.1 MODIFICATIONS INDISPENSABLES

Calcul de Z

Il serait important de tenir compte des interlignes standards dans le calcul de Z.

$$Z = (n-1)*I + \sum_{i=1}^n h_{itot} ,$$

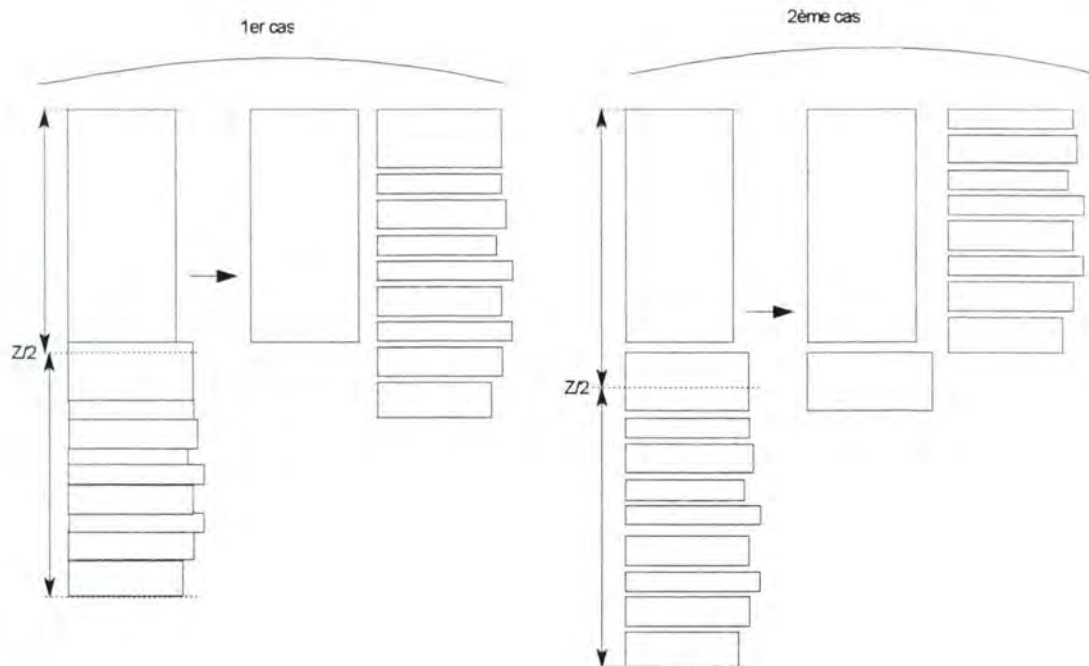
avec I = Interligne standard,
 h_{itot} = hauteur totale de l'élément i .

Sinon, dans le cas où l'on aurait un gros élément d'une hauteur h_1 suivi de $n-1$ petits

éléments de hauteur h_i avec $\sum_{i=2}^n h_i \approx h_1$, cela conduirait à $Z = h_1 + \sum_{i=2}^n h_i$. La division de Z par 2 placerait le 1^{er} élément dans la 1^{ère} colonne et les autres éléments dans la 2^{ème} colonne (2^{ème} cas de la figure ci-dessous). Alors qu'une meilleure répartition aurait certainement conduit à placer le 1^{er} des n petits éléments dans la 1^{ère} colonne (2^{ème} cas de la figure ci-dessous).

Après insertion des interlignes, la hauteur finale de la 2^{ème} colonne devient :

$$(n-1)*I + \sum_{i=2}^n h_i .$$



Envisager la division de Z par 2 lorsque :

* L_{\max} ne dépasse pas la moitié de l'écran

* et $(Z / L_{\max}) > 0,9$,

où L_{\max} = longueur maximum de tous les éléments.

Justification des boîtes de regroupement

Lorsqu'on a une boîte de regroupement de niveau 1 (la boîte n'est pas contenue dans une boîte de regroupement), les objets à l'extérieur qui ne sont pas des boîtes de regroupement, doivent être décalés de L (intercolonne standard) vers la droite par rapport à la marge initiale qui a été fixée.

Dans le cas d'une boîte de regroupement de niveau n ($2 \leq n \leq 4$, pour éviter la saturation visuelle), son cadre doit être décalé de $(n-1)*L$ unités vers la droite par rapport à la marge initiale fixée.

Les objets de niveaux $n-1$ doivent être décalés de $n*L$ unités vers la droite par rapport à la marge initiale fixée.

Les boîtes de regroupement de même niveau et de la même colonne doivent être uniformisées à condition qu'elles aient des largeurs similaires.

Justification des boutons

Il serait préférable de justifier à droite les boutons de commande dans le cas où l'on aurait choisi de les disposer horizontalement. Si le nombre de boutons est suffisant, il faudrait les distribuer de façon à remplir toute la largeur.

Modification de l'heuristique de placement des boutons de commande

si $1 \leq p_{\text{int}} \leq 1,6$

alors si $h_{\text{vert}} \leq h_{\text{tot}}$

alors disposer le groupe de boutons verticalement à droite (cas 1)

sinon si $l_{\text{horiz}} \leq l_{\text{tot}}$

alors disposer les boutons horizontalement en bas (cas 2)

sinon si $g \neq 0$
 alors disposer les boutons en 2 groupes
 verticaux à droite (cas 3)
 sinon disposer les boutons verticalement
 à droite (cas 4)

sinon si $p_{int} < 1$
 alors disposer les boutons verticalement à droite (cas 5)
 sinon disposer les boutons horizontalement en bas (cas 2)

avec $p_{int} = (l_{tot} / h_{tot})$,
 l_{tot} = la longueur totale des éléments placés,
 h_{tot} = la hauteur totale des éléments placés,
 h_{vert} = la hauteur totale des boutons placés verticalement,
 l_{vert} = la longueur maximum des boutons placés verticalement.

IV.5.2 QUELQUES MODIFICATIONS PRÉFÉRABLES

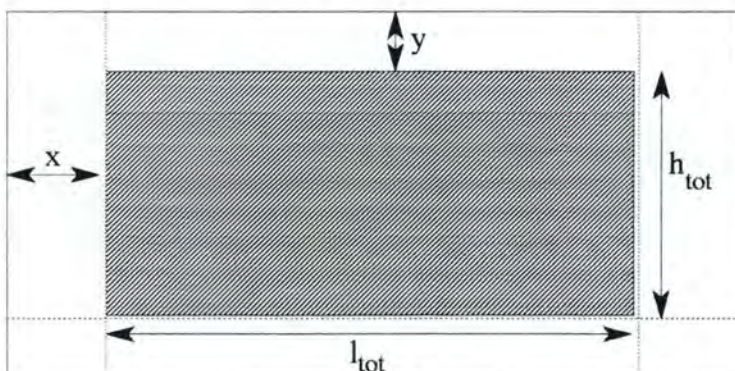
Isolation des boutons de commande

Laisser 4L entre les boutons et les autres objets lors d'une disposition verticale des boutons et 2I dans le cas d'une disposition horizontale. Ceci augmenterait l'isolation des boutons de commande.

Calcul des marges d'encadrement

La proposition d'heuristique faite pour le choix des marges nous conduit toujours à la situation où les marges latérales sont supérieures ou égales aux marges supérieures et inférieures. Or, nous devons éviter d'avoir les marges latérales supérieures aux marges supérieures et inférieures.

Supposons qu'après placement des différents OIC nous ayons la forme suivante qui est hachurée, et déterminons les distances x et y correspondant respectivement aux marges latérales et aux marges supérieures et inférieures :



D'après l'algorithme, $\exists m \in M$ tel que : $p_{int} \leq m = \min_x \{x \in M : x \geq p_{int}\}$.

Nous avons donc,

$$(l_{tot} / h_{tot}) \leq m \quad (1),$$

et nous cherchons à obtenir un

$$p_{\text{intfinal}} = (l_{\text{tot}} + 2x) / (h_{\text{tot}} + 2y) = m \quad (2).$$

$$(1) \text{ et } (2) \Rightarrow l_{\text{tot}}(h_{\text{tot}} + 2y) \leq (l_{\text{tot}} + 2x)h_{\text{tot}} \quad (3).$$

$$(3) \Rightarrow l_{\text{tot}} * 2y \leq 2x * h_{\text{tot}} \quad (4).$$

$$(4) \Rightarrow (y/x) \leq (h_{\text{tot}} / l_{\text{tot}}) \quad (5).$$

Nous savons d'autre part que l'interface finale doit vérifier,

$$(h_{\text{tot}} / l_{\text{tot}}) \leq 1 \quad (6). \text{ Autrement elle devrait être réorganisée.}$$

$$(5) \text{ et } (6) \Rightarrow y \leq x.$$

Or nous devons éviter les cas où $y < x$.

Proposition de modification

Il serait préférable de privilégier les deux proportions 1.6 et 1.33 qui sont les plus centrales dans le spectre des proportions fréquemment utilisées.

- a) si $p_{\text{int}} \leq 1,4142$ faire tendre le nouveau p_{int} vers **1,33** ;
si $p_{\text{int}} > 1,4142$ faire tendre le nouveau p_{int} vers **1,6**.
- b) Choisir une précision ϵ et un accroissement Δ sur les dimensions.

Par exemple, fixer au départ

$$\epsilon = 0,01 ,$$

$$\Delta = H/6,$$

$$\text{marge de droite ou de gauche : } x_i = 2L,$$

$$\text{marge de haut ou de bas : } y_i = H.$$

Supposons que l'on veuille atteindre la proportion interne 1,33.

Calculer le nouveau p_{int} .

Si $|p_{\text{int}} - 1,33| = \epsilon_i \leq \epsilon$, alors s'arrêter.

Sinon, faire croître y_i de Δ et réitérer le processus de calcul de p_{int} . Le comparer à 1,33, à ϵ près.

Poursuivre l'itération tant que la précision désirée n'est pas atteinte. Si au bout de 4 accroissements suivis des tests on n'obtient toujours pas le résultat attendu, croître x_i de L . Au bout de 4 accroissements de x_i , si on n'a toujours pas la précision voulue, arrêter les calculs et choisir l'itération ayant donné le plus petit ϵ_i .

IV.5.3 AUTRES REMARQUES ET SUGGESTIONS

Heuristique de justification pour des éléments semblables

Eléments semblables du point de vue de leurs formes :

SBX (liste de combinaison déroulante),

EBX (champ d'édition unilinéaire),

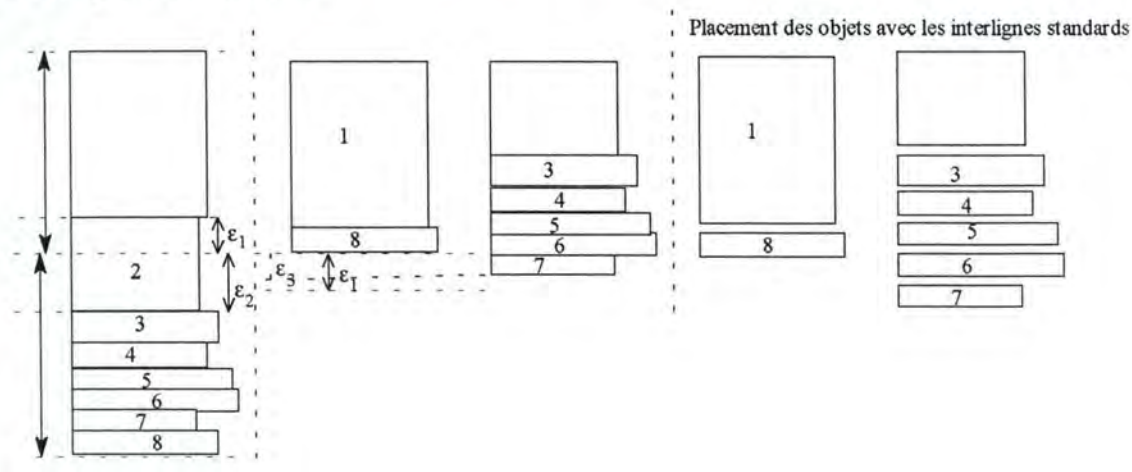
MBX (champ d'édition multilinéaire) ,

SLB (liste de sélection déroulante).

Utiliser les mêmes justifications au cas où 2 éléments semblables se suivraient.

Transfert éventuel du dernier OIC de droite à gauche

Le transfert du dernier élément de la deuxième colonne vers la première colonne peut entraîner une rupture dans l'ordre logique des objets interactifs (voir figure ci-dessous). Dans le cas de l'exemple traité ci-dessus, *Régime* pourrait venir après la boîte de regroupement *Patient*. Ce problème de transfert ne se serait certainement pas posé si l'on avait fait intervenir les interlignes standards dans le calcul de $Z/2$. Cela aurait permis de placer le deuxième élément dans la première colonne.



Remarques sur les boutons-radio

1°) Nous partons de l'hypothèse que le nombre d'items de bouton-radio est **inférieur ou égal à 4** dans un même bouton. L'important dans ce principe n'est pas de limiter à 4 le nombre d'items dans un bouton-radio, mais surtout d'éviter un affichage trop chargé si le nombre d'items est élevé. Ainsi on pourrait accepter :

- un bouton-radio à 4 "grands" items,

— Carte de Crédit —

☒ Visa Pension

☐ Master Card

☐ American Express

☐ Carte Bleue

- un bouton-radio à 6 petits items,

— Système —

☒ Dos ☐ OS/2 ☐ Top view

☐ Windows ☐ GEM ☐ X

Mais il vaut mieux s'en tenir à une règle stable.

2°) Le placement des items de boutons-radio est figé. Ce qui a pour inconvénient une perte de place et un manque de flexibilité dans les dispositions.

Mais cela a pour avantage d'être cohérent et de respecter un ordre de lecture verticale.

Remarques sur les boutons de commande

L'espace libre en dessous de la 2^{ème} colonne n'est pas considéré pour le placement des boutons de commande (voir surface hachurée de la figure ci-dessous).

Admission

Date du jour :

Patient

Nom :

Prénom :

Date de Naissance :

Adresse :

Téléphone :

Sexe

☒ Masculin

☐ Féminin

Etat civil

☒ Célibataire

☐ Marié

☐ Veuf

☐ Divorcé

Code de l'organisme :

Numéro de Titulaire :

Type d'affiliation :

Médecin :

Service : ▼

Catégorie de chambre

☒ Chambre particulière

☐ Chambre à 2 lits

☐ Chambre à 4 lits

Régime : ▼

Ok

Annuler

Avantage :

Tout est cohérent et facilite l'apprentissage et la lecture. De même l'algorithme se trouve ainsi simplifié.

Inconvénients :

- 1°) L'isolation n'est pas suffisant. Pour palier cela, il serait préférable de prévoir la possibilité d'ajouter séparateur vertical (cas d'un placement des boutons de commande à droite) ou d'un séparateur horizontal (cas d'un placement des boutons en bas).
- 2°) L'espace blanc non utilisé en dessous de la 2^{ème} colonne n'est pas pris en compte. Une possibilité de résorber cet espace est d'essayer après, d'y placer si cela est possible un ou plusieurs petits objets manuellement.
- 3°) Allongement indésirable des dimensions (cas par exemple d'un bouton avec un long libellé tel que : Réinitialiser les données...). Il serait mieux de refuser les longs libellés (ne pas dépasser par exemple 15 lettres pour un libellé) ; sinon il faudrait le placer manuellement.

Placement des objets interactifs concrets dans TRIDENT

91

IV.5.4 STRATEGIE "DROITE-DESSOUS" A UN NIVEAU

Nous estimons que l'imposition d'une disposition en deux colonnes est une très forte contrainte. Il n'est pas toujours nécessaire en effet de faire une répartition en deux colonnes lorsque par exemple, le nombre d'éléments à placer est petit.

Nous ne retenons plus l'hypothèse de la disposition des OIC dans deux colonnes.

En partant d'une situation notée (s_i), la situation notée ($s_{i,1}$) est obtenue en plaçant l'OIC suivant à droite de celui précédemment placé et celle notée ($s_{i,2}$) est obtenue en plaçant l'OIC suivant en dessous du premier OIC de la ligne précédente (nous proposons une illustration de la stratégie "Droite-Dessous" à la fin de cette section).

Localisation des OIC

1°) Placement à droite

Pour que le placement à droite soit possible il faut que la longueur totale n'excède pas la limite permise, c'est-à-dire :

$$L_T = \sum_{i=1}^n l_i + lb_{\max} + [2(n-1)+4]L < l_e,$$

avec n = nombre d'OIC sur une ligne sauf les boutons de commande,

l_i = longueur du $i^{\text{ème}}$ OIC,

lb_{\max} = longueur maximum des boutons,

L = inter colonne standard ($\frac{1}{2} U$ dans notre exemple),

l_e = largeur de l'écran.

La stratégie "Droite-Dessous" doit aussi assurer une continuité visuelle des objets interactifs.

Trois cas sont à envisager dans le placement à droite :

a) L'élément placé à gauche et celui à placer à droite sont de même hauteur.

Dans ce cas le placement à droite se fait avec alignement par rapport à l'élément précédent.



b) L'élément à placer à droite est de hauteur plus petite que celle de l'élément précédent.

si l'élément à placer est un champ d'édition

alors si l'élément précédent est une liste de sélection ou un champ d'édition

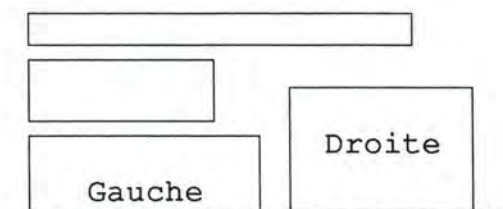
alors aligner les deux libellés.

libellé1: ▼ libellé2:

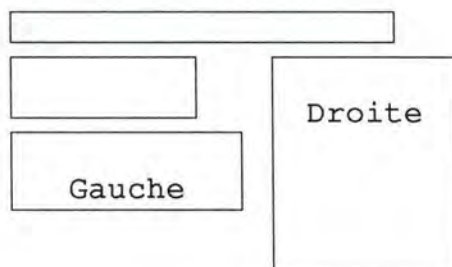
sinon placer l'élément en alignant son bord supérieur par rapport à celui de l'élément précédent. (On privilégie le fait que l'on puisse utiliser par la suite l'espace en dessous du dernier élément placé)



- c) L'élément à placer à droite est de hauteur plus grande que celle de l'élément précédent.
si l'espace disponible permet de placer l'élément à droite en alignant les bords inférieurs
alors placer l'élément avec alignement des bords inférieurs



sinon le placer le plus haut possible (l'élément placé occupe l'espace inutilisé au dessus et déborde le bord inférieur de l'élément précédent)



2°) Placement en dessous

Le placement en dessous est élémentaire et consiste à placer l'OIC en dessous du premier OIC de la ligne précédente avec justification des bords gauches.

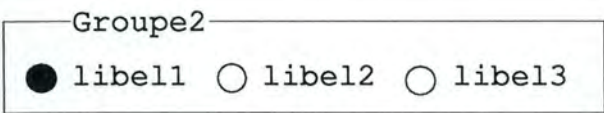
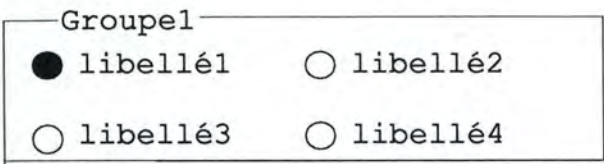
IV.5.5 ARRANGEMENT DES ITEMS DE BOUTONS-RADIO

Dans le cas où l'on aurait plus d'un bouton-radio de même niveau il est nécessaire de commencer par décider de la disposition des items dans le bouton contenant le plus d'items.

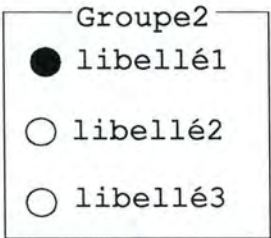
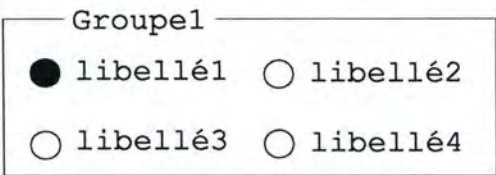
- 1°) Dans un bouton de 4 items, ils doivent être disposés soit sur 2 lignes de 2 items chacune, soit tous les 4 verticalement si les libellés sont très longs.

2°) Dans un groupe de 3 items

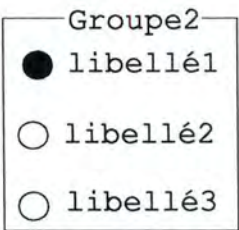
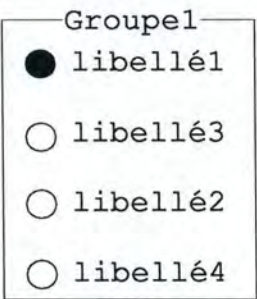
- a) Il existe un autre groupe de 4 items.
si il a été décidé de disposer les items du bouton de 4 items en 2 lignes de 2 items
alors si les longueurs des libellés permettent une disposition horizontale de tous les items dans le bouton de 3 items (voir remarque ci-dessous)
alors choisir une disposition horizontale de tous les items



sinon choisir une disposition verticale de tous les 3 items.



sinon si il a été décidé de disposer verticalement tous les 4 items
alors choisir une disposition verticale des items du bouton de 3.



- b) Le bouton de 3 items est celui qui dispose du maximum de items
si les longueurs des libellés permettent une disposition horizontale (voir remarque ci-dessous)
alors disposer horizontalement tous les 3 items
sinon disposer tous les 3 verticalement.

Remarque :

On dira que les longueurs des libellés permettent une disposition horizontale si la longueur de l'item le plus long multipliée par le nombre d'items de bouton-radio que l'on désire mettre sur une ligne n'est pas supérieure à L_{\max} . L_{\max} est calculée itérativement et est égale à la longueur maximum de tous les éléments déjà considérés.

3°) Dans un bouton à 2 items

- a) Il existe un autre bouton de 3 ou 4 items
si il a été décidé une disposition horizontale pour un bouton ayant plus de 2 items
alors si les longueurs des libellés permettent une disposition horizontale
alors choisir une disposition horizontale

Groupe1

<input checked="" type="radio"/> libell1	<input type="radio"/> libell2	<input type="radio"/> libell3
--	-------------------------------	-------------------------------

Groupe2

<input checked="" type="radio"/> libellé1	<input type="radio"/> libellé2
---	--------------------------------

sinon choisir une disposition verticale

Groupe2

<input checked="" type="radio"/> libellé1
<input type="radio"/> libellé3

- sinon choisir une disposition verticale
b) Il n'existe pas de bouton de plus de 2 items
si les longueurs des libellés permettent une disposition horizontale
alors choisir une disposition horizontale des 2 items

Groupe1

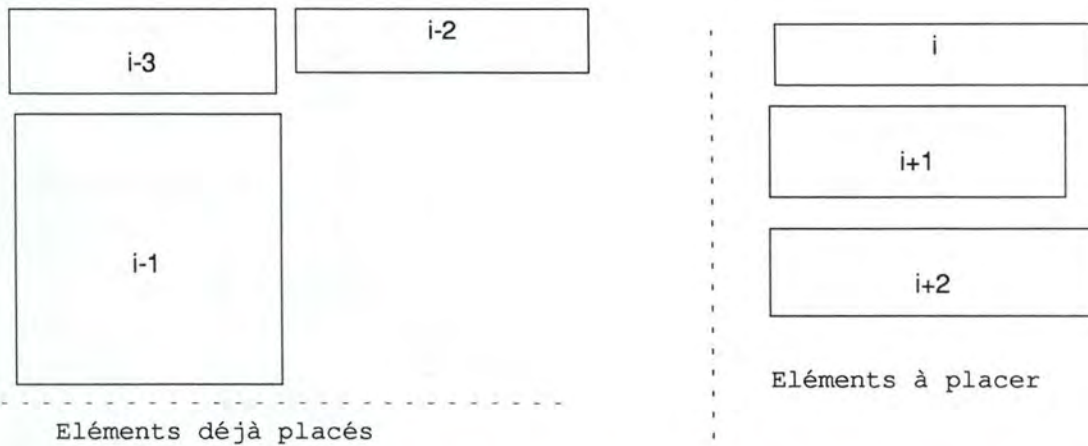
<input checked="" type="radio"/> libellé1	<input type="radio"/> libellé2
---	--------------------------------

sinon choisir une disposition verticale

Remarque :

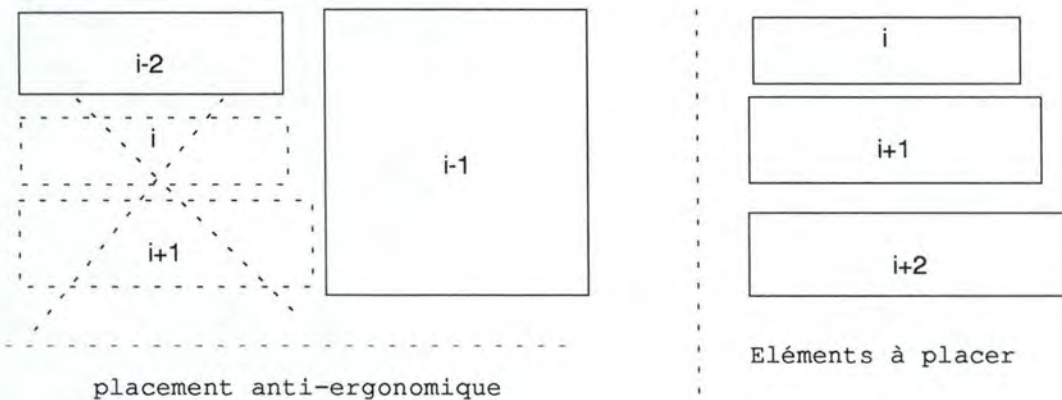
Dans le cas où les dimensions des éléments sont disproportionnées (des gros éléments et des petits éléments à placer) il faut étendre la stratégie à plusieurs niveaux

a) Situation notée s_i



Si l'espace en dessous de $i-2$ est grand, placer l'élément i sous $i-2$. Appliquer le même procédé pour $i+1$ et les suivants tant que la condition reste encore vérifiée.

b) situation notée s_i



Il faut éviter de placer les éléments $i, i+1, \dots$ en dessous de $i-2$.

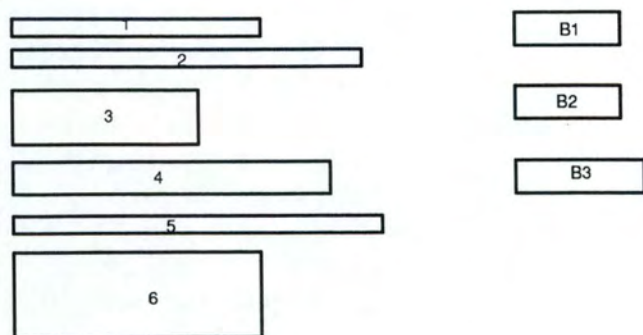
Nous illustrons la stratégie Droite-Dessous à un niveau sur un petit exemple.

Hypothèse : Les boutons de commande sont placés verticalement à droite.

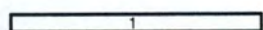
Largeur de l'écran : 33 unités, hauteur de l'écran : 20 unités ; ces dimensions ont été choisies dans le souci de respecter les proportions dans la réalité.

Dimensions des OIC à placer :

- 1 : (8 ; 0,5)
- 2 : (12 ; 0,5)
- 3 : (6 ; 1,5)
- 4 : (10 ; 1)
- 5 : (13 ; 0,5)
- 6 : (8 ; 2,5)
- B1 : (3 ; 1)
- B2 : (3 ; 1)
- B3 : (4 ; 1)



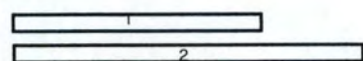
(1)



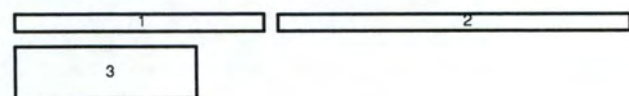
(1.1)



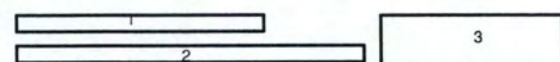
(1.2)



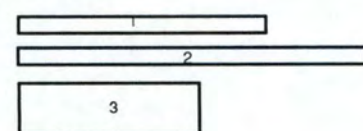
(1.1.1)



(1.1.2)

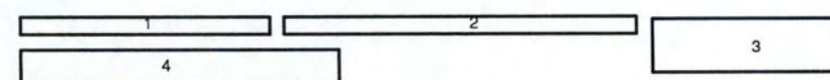


(1.2.2)

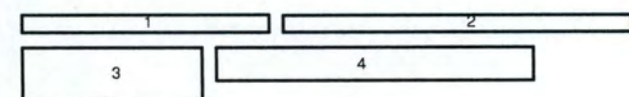


(1.1.1.1) $L_T > 33$.

(1.1.1.2)



(1.1.2.1)

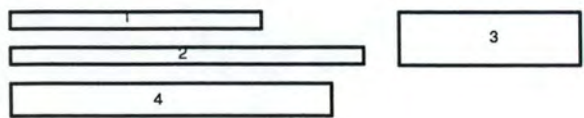


(1.1.2.2)

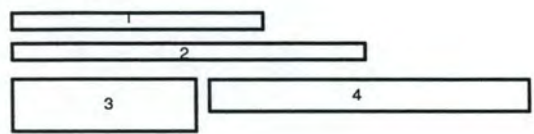


(1.2.1.1) $L_T > 33$.

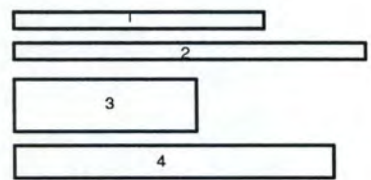
(1.2.1.2)



(1.2.2.1)

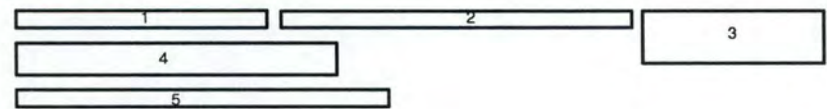


(1.2.2.2)



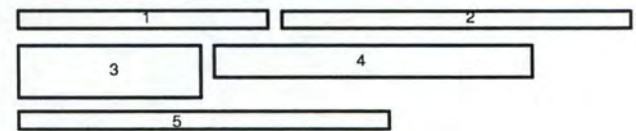
(1.1.1.2.1) Impossible : l'élément 5 ne peut s'insérer entre les éléments 3 et 4.

(1.1.1.2.2)



(1.1.2.1.1) $L_T > 33$.

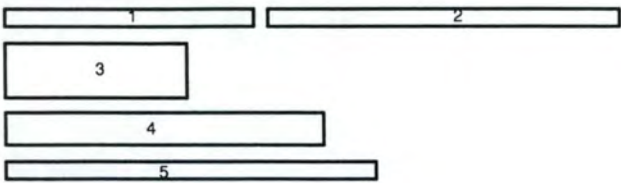
(1.1.2.1.2)



(1.1.2.2.1)



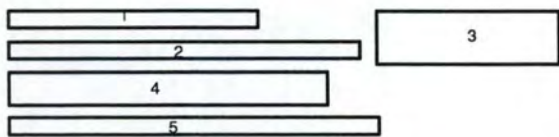
(1.1.2.2.2)



(1.2.1.2.1)

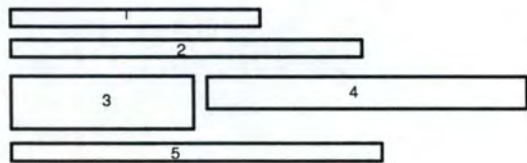


(1.2.1.2.2)

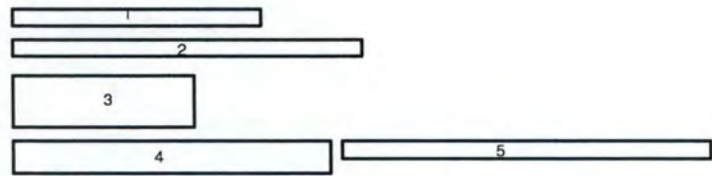


(1.2.2.1.1) $L_T > 33$.

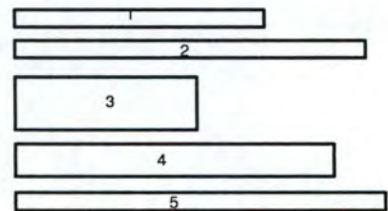
(1.2.2.1.2)



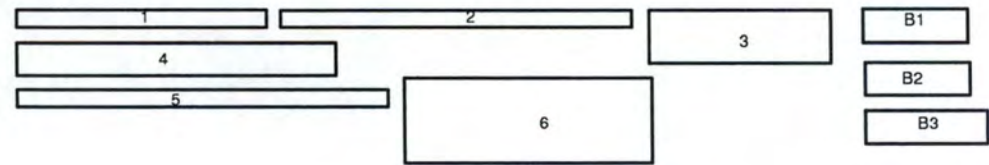
(1.2.2.2.1)



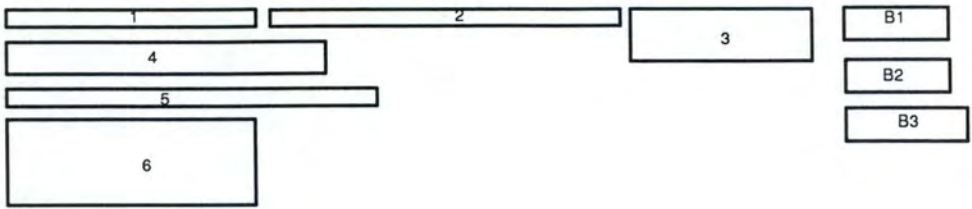
(1.2.2.2.2)



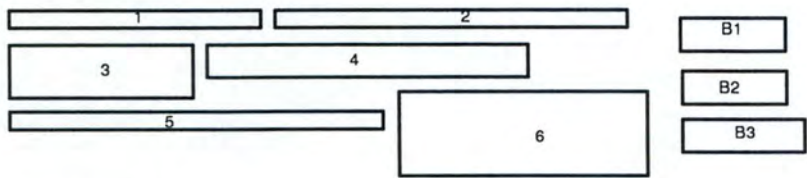
(1.1.1.2.2.1)



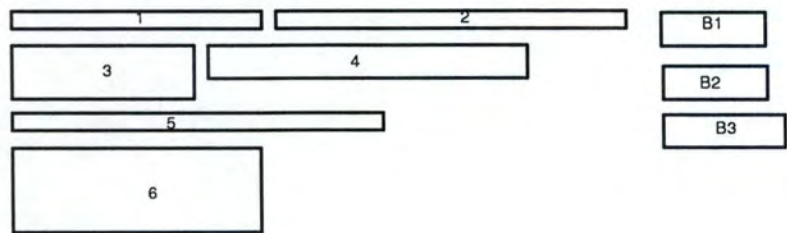
(1.1.1.2.2.2)



(1.1.2.1.2.1)

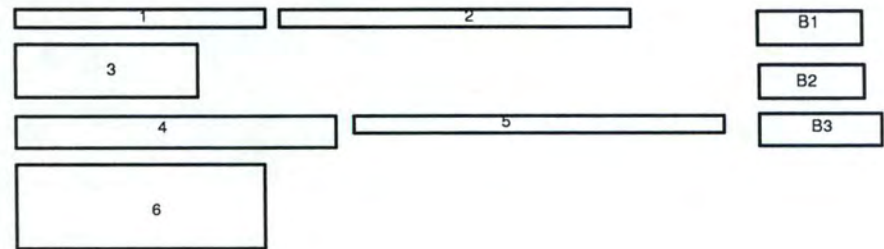


(1.1.2.1.2.2)

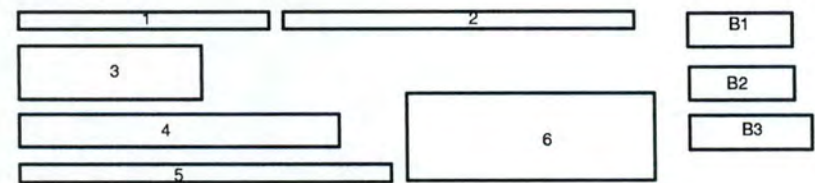


(1.1.2.2.1.1) $L_T > 33$.

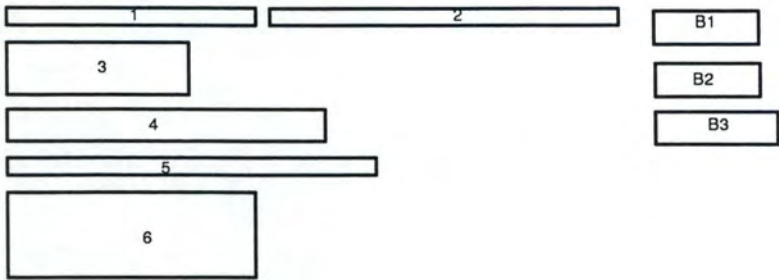
(1.1.2.2.1.2)



(1.1.2.2.2.1)

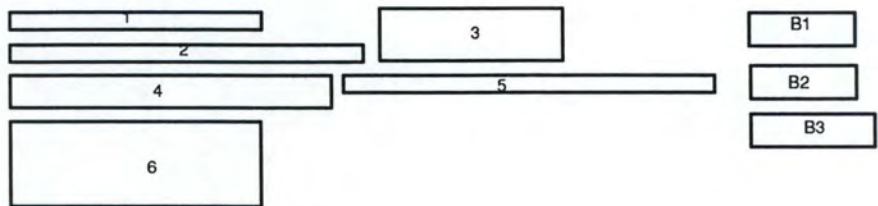


(1.1.2.2.2)

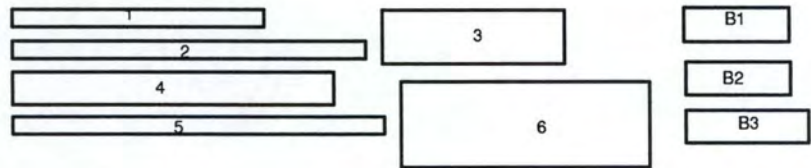


(1.2.1.2.1.1) $L_T > 33$.

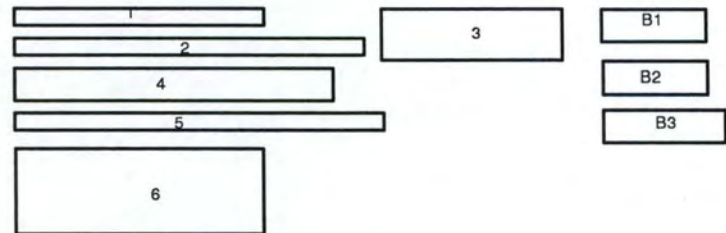
(1.2.1.2.1.2)



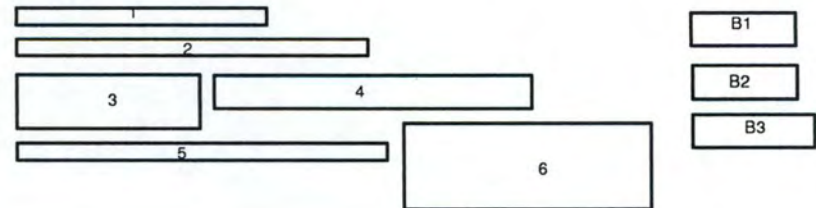
(1.2.1.2.2.1)



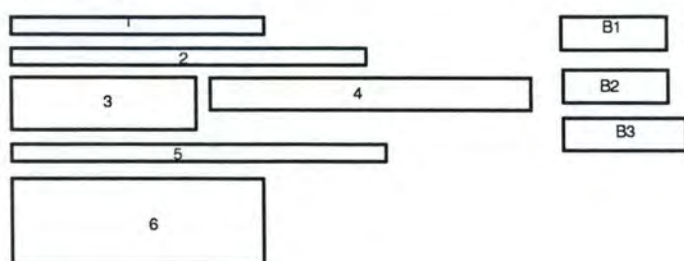
(1.2.1.2.2.2)



(1.2.2.1.2.1)

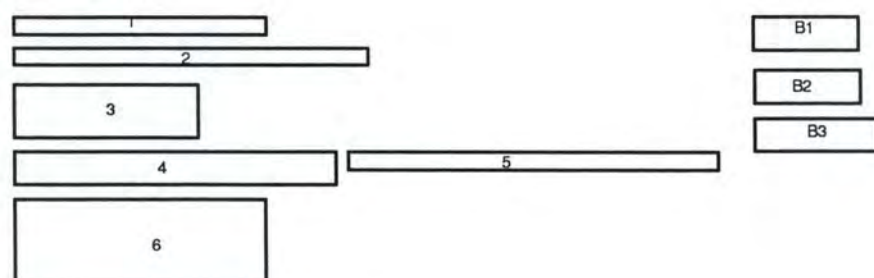


(1.2.2.1.2.2)



(1.2.2.2.1.1) $L_T > 33$.

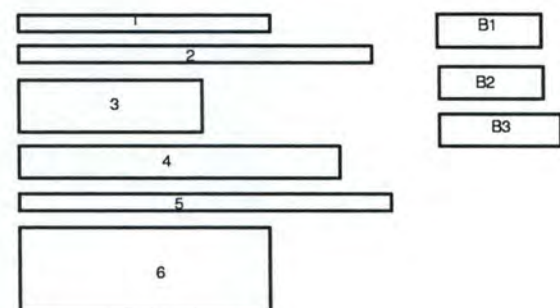
(1.2.2.2.1.2)



(1.2.2.2.2.1)



(1.2.2.2.2.2)



La démarche reste la même lorsque l'on décide de placer les boutons de commande horizontalement en bas. Seule différence, la formule qui limite L_T devient moins contraignante.

Soit n le nombre d'OIC à placer (nous ne prenons pas en compte les boutons de commande). Le nombre maximal d'alternatives auxquelles conduit la stratégie Droite-Dessous à un niveau avec boutons de commande à droite est 2^{n-1} . Dans l'exemple traité, ce nombre est 32. En ajoutant les contraintes d'espace, ce nombre diminue considérablement : ici, il est passé de 32 à 15. D'autres contraintes peuvent être ajoutées

pour éliminer les alternatives qui n'ont aucune chance d'être retenues ; par exemple celle donnée plus haut pour l'arrangement des items de boutons-radio.

Après la génération, le concepteur peut raffiner par manipulation directe l'interface de son choix, essentiellement pour améliorer l'alignement et la justification des OIC. Une autre possibilité est d'ajouter des heuristiques à la stratégie pour que ces alignements et justifications se fassent de façon automatique.

IV.6 CONCLUSION

Dans le but d'améliorer l'algorithme TRIDENT, un certain nombre de critiques ont été portées à cet algorithme et des propositions ont été faites. Dans un souci de clarté, des exemples et des dessins étayent certaines propositions.

Nous avons ajouté une stratégie appelée "Droite-Dessous" afin, entre autres, de :

- minimiser les espaces libres non utilisés et en veillant à respecter la succession logique des OIC,
- briser la contrainte du bi-colonnage,
- conférer une plus grande flexibilité de placement,
- rendre la stratégie "plus intelligente" par l'adjonction de nouvelles heuristiques à chaque étape,
- promouvoir une image informationnelle plus continue et compacte,
- respecter l'ordre des OIA,
- être plus souple à chaque étape tout en restant "assez simple".

Compte tenu de la difficulté de concevoir des interfaces répondant au goût de tous les utilisateurs, il serait plus judicieux de leur offrir la possibilité de raffiner itérativement l'interface générée automatiquement.

CHAPITRE V

CONCLUSION GENERALE

De nos jours, les informaticiens accordent de plus en plus une place importante au problème des Interfaces Homme-Machine dans la réalisation de leurs applications. Un logiciel ne peut avoir du succès sur le marché que si son interface est attrayante et offre une facilité d'apprentissage aux utilisateurs. Savoir donc où placer sur l'écran, un objet de l'interface qui est amené à être manipulé par l'utilisateur, tout en tenant compte de ses habitudes et en évitant de lui demander des efforts inutiles, devient l'une des tâches prioritaires à accomplir dans la réalisation d'un logiciel. Notre travail s'est intéressé à ce problème de placement des objets interactifs suivant les trois aspects : localisation, dimensionnement et arrangement.

Le but visé à travers cette étude était :

- 1°) d'examiner les outils existants qui traitent de ce problème et les solutions proposées afin de pouvoir en tirer des enseignements nécessaires ;
- 2°) de proposer une stratégie de placement efficace et systématique à partir de la connaissance des règles ergonomiques et des leçons tirées de l'étude des outils existants. Nos propositions devaient tenir compte des coûts que cela pourrait occasionner.

Les solutions proposées tentent de résoudre entre autres les problèmes des espaces libres, des boutons-radio, des boutons de commandes. Les solutions exigeant un coût prohibitif (par exemple les post-justifications qui demandent un réexamen total à chaque étape) n'ont pas été retenues.

- *Problème de minimisation des espaces libres*

L'étude sur les outils existants nous a permis de faire certaines propositions en tenant compte des avantages et des inconvénients de certaines solutions rencontrées au cours de cette étude. C'est le cas par exemple de la stratégie "Droite-Dessous" que nous avons proposée et qui, entre autres, minimise les espaces libres non utilisés en respectant la succession logique des items. L'algorithme DON minimise aussi les espaces libres non utilisés. Il ne tient cependant pas compte de la succession logique des OIC lorsqu'ils sont considérés comme feuilles d'un même nœud.

- *Problème des boutons-radio*

Les propositions faites sur les arrangements des items des boutons-radio permettent aussi non seulement de minimiser les espaces libres mais aussi de garder une cohérence dans leurs dispositions.

- *Problème des boutons de commandes*

La solution de disposer les boutons de commande soit verticalement à droite, soit horizontalement en bas de la boîte de dialogue a pour objectif de mieux les isoler et de les garder à des places fixes. Cela facilite l'apprentissage de l'utilisateur.

- *Problème d'ergonomie*

Les règles ergonomiques n'ont pas été négligées dans cette étude. En effet, les aspects tels que la cohérence spatiale qui peut faciliter l'apprentissage de l'utilisateur, l'arrangement pour renfort visuel, l'arrangement esthétique ont été pris en compte. Les propositions d'heuristiques sur les arrangements des boutons-radio, l'isolation des boutons de commande en sont des exemples illustratifs.

L'application de l'algorithme TRIDENT à un exemple nous a permis de mettre en évidence certaines insuffisances et donc de faire des propositions en conséquence. C'est le cas par exemple, de la justification des formes semblables entre elles, de la justification et de la disposition des boîtes de regroupement de n'importe quel niveau ainsi que de la disposition des éléments à l'extérieur d'une boîte de regroupement.

Pour résoudre le problème de flexibilité, il serait préférable d'offrir la possibilité à l'utilisateur de raffiner son interface par manipulation directe car il est difficile sinon impossible de générer automatiquement une interface répondant à **tous** les goûts et respectant **toutes** les règles ergonomiques. D'autant plus que dans la pratique certaines règles ergonomiques peuvent être prioritaires par rapport à d'autres lorsque ces règles entre en conflit.

L'étendue de notre travail ne nous pas permis d'envisager le problème de l'implémentation qui devrait être la prochaine étape.

BIBLIOGRAPHIE

- [DEB92] De Baar D., Foley J. D., et Mullet K. E., "*Coupling Application Design and User Interface Design*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'92, (Monterey, 3-7 mai 1992), pp. 259-266.
- [FOL92] Foley J. D., Kim W. C., Kovacevic S., et Murray K., "*UIDE - An Intelligent User Interface Design Environment*", in J. Sullivan and S. Tyler (eds.), *Intelligent User Interfaces*, ACM Press, 1992, pp. 339-384
- [JAN93] Janssen C., Weisbecker A. et Ziegler J., "*Generating User Interfaces from Data Models and Dialogue Net Specification*", in Proceedings of the Conference on Human Factors in Computing Systems, INTERCHI'93, (Amsterdam, 24-29 avril 1993), ACM Press, New York, 1993, pp. 418-423.
- [JOH92] Johnson, J., "*Selectors : Going beyond User-Interface Widgets*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'92, (Monterey, 3-7 mai 1992), pp. 273-279.
- [KIM90] Kim, W. C. et Foley, J. D., "*DON : User Interface Presentation Design Assistant*", in Proceedings of the Symposium on User Interface Software and Technology, UIST'90, (Snowbird, octobre 1990), pp. 10-20.
- [KIM93] Kim, W. C., "*Providing High-Level Control and Expert Assistance in the User Interface Presentation Design*", in Proceedings of INTERCHI'93, (Amsterdam, 24-29 avril 1993), ACM Press, New York, 1993, pp. 430-437.
- [LAR92] Larson, J.A., "*Interactive Software : Tools for Building Interactive User Interfaces*", Englewood Cliffs, NY : Yourdan Press, 1992, pp. 298-300.
- [MEI91] Meinadier, J. P., "*L'Interface Utilisateur : Pour une Informatique plus Conviviale*", Dunod, Paris, 1991.
- [PET89] Petoud, I. et Pigneur, Y., "*An Automatic and Visual Approach for User Interface Design*" in IFIP Working Conference Engineering for Human Computer Interaction, (Napa Valley, août 1989), pp. 7-14.
- [PET90] Petoud, I. "*Génération Automatique de l'Interface Homme-Machine d'une Application de Gestion hautement interactive*", Inf. Chabloz, Tolochenaz, 1990, pp.171-192.
- [SAC92] Sacré, B., Sacré-Provot, I. et Vanderdonckt, J. "*Une Description Orientée Objet des Objets Interactifs Abstraits utilisés en Interface Homme-Machine*", FUNDP, Institut d'Informatique, Namur, octobre 1992.
- [SZE92] Szekeley P., Luo P. et Neches R., "*Facilitating the Exploration of Interface Design Alternatives : The HUMANOID Model of Interface Design*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'92, (Monterey, 3-7 mai 1992), pp. 507-515.

- [SZE93] Szekeley P., Luo P. et Neches R., "*Beyond Interface Builders : Model-Based Interface Tools*", in Proceedings of the Conference on Human Factors in Computing Systems INTERCHI'93, (Amsterdam, 24-29 avril 1993), pp. 383-390.
- [VAN90] Vander Zanden B., Myers, B. A., "*Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'90, (Seattle, avril 1990), pp. 27-34.
- [VAN93a] Vanderdonckt, J. "*Guide Ergonomique de la Présentation des Applications Hautement Interactives*", FUNDP, Institut d'Informatique, Namur, février 1993.
- [VAN93b] Vanderdonckt, J. et Bodart, F. "*Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection*", in Proceedings of INTERCHI'93, (Amsterdam, 24-29 avril 1993), ACM Press, New York, 1993, pp. 424-429.
- [WIE89] Wiecha C., Bennett W., Boies S., Gould J., "*Generating Highly Interactive User Interfaces*", in Proceedings of the Conference on Human Factors in Computing Systems, CHI'89 (Austren, may 1989), pp. 277-282.